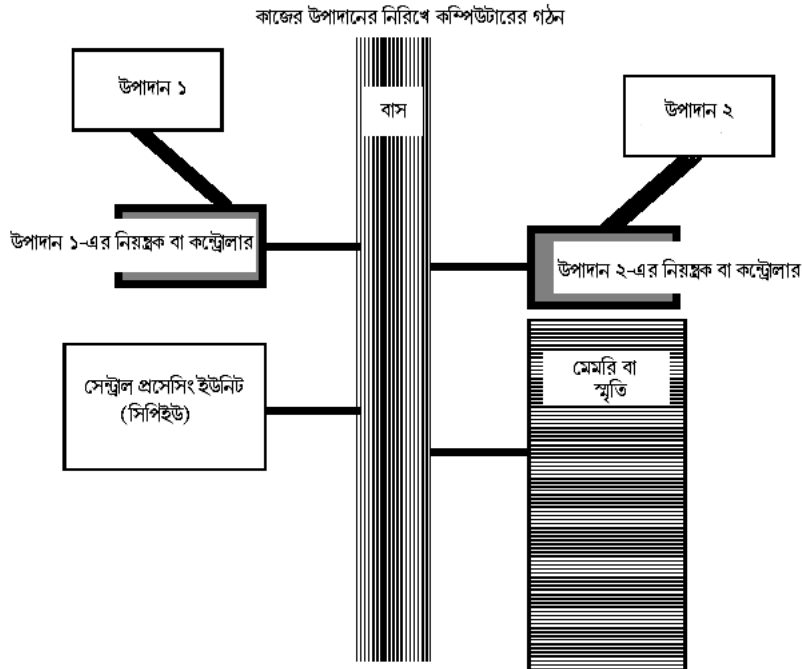
জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

আমরা শূন্যতা থেকে বেরিয়ে এসেছি। এখন আমাদের সামনে পিসিটা মানে কম্পিউটারটা ইতিমধ্যেই সদাসর্বদাই, অলওয়েজ অলরেডি, আছে। আমরা এবার কম্পিউটারের কাজ করার প্রক্রিয়ার আলোচনায় ঢুকব। আজই আদতে আমাদের আলোচনার প্রথম দিন — দিন নাম্বার ওয়ান — নাম্বার ওয়ান সিরিজের প্রভু জয় গোবিন্দা জয় ডেভিড বলে শুরু করা যাক। শূন্যতম দিনে সরাসরি কোনো বইয়ের রেফারেন্স দেওয়া হয়নি, রেফারিত হওয়ার প্রয়োজন পড়বে তারই যে এই এক অব্দি আসছে। দিন শূন্য-র মূল রেফারেন্স 'পিসি হার্ডওয়ার ইন এ নাটশেল' আর পিটার নর্টনের 'ইনসাইড ইয়োর পিসি'। এর বাইরে থেকে অনেক আলোচনা এসেছে — সেগুলো বহুমুখী বিচিত্রমুখী। ছবিগুলো নামানো হয়েছে মূলত নেট থেকে, কিছু বিভিন্ন এনসাইক্লোপিডিয়া থেকে, এবং কিছু আমি, হুঁ হুঁ, সরাসরি নিজে এঁকেছি, তাও আবার বাঁ-হাতে মাউস ধরে। এবং নামানো ছবিগুলোর নকল করার অধিকার, কপিরাইট, আইন বাঁচাতে, কারণে, এবং অকারণেও, ছবির শরীরে বহু লেখা গুঁজে দেওয়া হয়েছে। অন্নু মালিকের মডেলে, এর অনু পরমানুর মালিক এখন আমরাই। এফএসএফ-এর সঙ্কর্ষণ ইতিমধ্যেই আমায় থ্রেট করেছে, দেখো, সাইটে যেন তোলা যায়। আমরা শূন্যয় কম্পিউটারের ভৌত উপাদানগুলোর কাজের ছক দেখেছিলাম। এবার শুরু করব কাজের ভিত্তিতে উপাদানগুলোর ছক দিয়ে। শুধু একটা কথা মাথায় রাখবেন, কোনো সেকশনেরই পুরোটা বোঝা যায়না অন্য সমস্ত সেকশনগুলো জেনে ফেলার আগে, আর পুরোটা বোঝা বলে কিছু হয়ই না। তাই, কোথাও, খুব ক্যাঁচাল বলে মনে হলে দাঁতে হাত রেখে পড়ে যাবেন, পরে দেখবেন এই করতে করতে কবে মাড়ি হয়ে উঠেছে শক্ততর, দাঁত পড়ে যাওয়ার কোনো ভয়ই নেই আর, বরং মনে হবে আরে, এটুকু দাঁত ও মাড়ির ব্যায়াম তো প্রত্যহ পালনীয়। আমরা একটু সিপিইউর কাজের রকম নিয়ে কথা বলব, তারপরেই চলে যাব অপারেটিং সিস্টেম ব্যাপারটা কী — সেটা খায় না মাথায় দেয় — সেই কথায়।

।। দিন এক।।

১।। কাজের কেন্দ্রীয় সংস্থা — সিপিইউ এবং মেমরি

একটা আধুনিক কম্পিউটারের মূল এবং প্রাথমিক অংশগুলো — এক বা একাধিক প্রসেসর, কিছু মূল মেমরি, ডিস্ক, প্রিন্টার, কিবোর্ড, মোডেম বা ল্যান-কার্ড জাতীয় নেটওয়ার্ক সংযোগের উপায়, এবং অন্যান্য ইনপুট-আউটপুটের মাধ্যম। সব মিলিয়ে একটা জটিল ব্যবস্থা। উপাদান হিশেবে আলাদা আলাদা অংশগুলোর আলাদা আলাদা ভূমিকা

এবং চরিত্র নিয়ে কিছু কথা আমরা শূন্য নম্বর দিনেই বলে এসেছি। কিন্তু সেখানে আমাদের মূল বোঝার ব্যাপারটা ছিল — একটা কম্পিউটারের যৌক্তিক গঠন। যুক্তিগত ভাবে কোন ভূমিকা কার। এবারে আমরা বুঝতে চেষ্টা করব বাস্তব ক্রিয়াটা ঘটার জন্যে কম্পিউটারের ভিতরকার আলাদা আলাদা উপাদানগুলো কী ভাবে পরস্পরের সঙ্গে যুক্ত থাকে। ছবিটা দেখুন — 'কাজের উপাদানের নিরিখে কম্পিউটারের গঠন'। স্মার্ট-কার্ড বা হ্যান্ডহেল্ড ডিভাইস ইত্যাদি খুব অন্যরকম মেশিনকে ছেড়ে দিলে অন্য প্রায় সবরকম কম্পিউটারের বেলাতেই এই ছবিতে দেখানো গঠনটা একটা অর্থ বহন করে। ছবিতে কেন্দ্রীয় জায়গায় আছে বাস বা যোগাযোগপথ। এই যোগাযোগপথই আলাদা আলাদা উপাদানগুলোর মধ্যে যোগাযোগ তৈরি রাখে যাতে তথ্যের চলাচল ঘটতে পারে কম্পিউটারের মধ্যে এক অংশ থেকে আর এক অংশে। শূন্য নম্বর দিনেই আমরা দেখলাম, কম্পিউটারের কাজের গোটা রসদটাই হল তথ্য। তথ্য কাজ করে আদেশ মোতাবেক। সেই আদেশগুলোও তথ্য।

কম্পিউটারটা যদি একটা সিস্টেম হয়, তার মধ্যে আবার অনেকগুলো সাবসিস্টেম আছে। সেই সিস্টেমাঙ্গদের ভিতর প্রমুখতম হল — সেন্ট্রাল প্রসেসিং ইউনিট বা সিপিইউ, মেমরি বা স্মৃতি, এবং পেরিফেরাল বা বহিরাঙ্গিক উপাদানগুলো, যেমন ফ্লপি ড্রাইভ, হার্ড ড্রাইভ ইত্যাদি। এই তিন সিস্টেমাঙ্গ মানে সিপিইউ, মেমরি এবং পেরিফেরাল — এরা প্রত্যেকেই অন্যের সঙ্গে সংযুক্ত থাকে বাস দিয়ে। বাস নিয়ে আলোচনা এখন বারবার আসবে। বাস মানে যোগাযোগ ব্যবস্থা। আমরা যেমন যোগাযোগের জন্যে, পথ অতিক্রম করার জন্যে, বাসে চড়ি। এই সাবসিস্টেমগুলো নিজেদের মধ্যে যোগাযোগ করে, তাদের মধ্যে তথ্য যাতায়াত করে, এই বাস দিয়ে। সবাই মিলে পরস্পর যোগাযোগের ভিত্তিতে কাজ করতে পারে এই বাসের মাধ্যমেই। এই ছবিতে দেখুন, দুটো পেরিফেরাল ডিভাইস বা বহিরাঙ্গিক উপাদানকে দেখিয়েছি আমরা — ১ আর ২। ধরুন ১ মানে কিবোর্ড, ২ মানে ফ্লপি, বা যে কোনো অন্য পেরিফেরাল। এই বহিরাঙ্গিক ডিভাইসরা তাদের ডিভাইস কন্ট্রোলারের মাধ্যমে যুক্ত থাকে এই বাস-সংযোগে। ডিভাইস বা উপাদান পিছু একটা করে ডিভাইস কন্ট্রোলার বা নিয়ন্ত্রক। বাসের সঙ্গে যুক্ত থাকে প্রতিটি ডিভাইসের কন্ট্রোলার, তার মারফত সেই ডিভাইসটা। ধরুন, সিপিইউ কোনো তথ্য পাঠাতে চাইছে কোনো একটা ডিভাইসকে, এই অবস্থায় সিপিইউ থেকে তথ্যটা প্রথমে আসবে বাসে, সেখান থেকে ডিভাইসের কন্ট্রোলারে, এবং সেখান থেকে যাবে ডিভাইসে। এই কন্ট্রোলারের কথায় আমরা একটু বাদেই আসছি।

কন্ট্রোলাররা যাদের কন্ট্রোল করে সেই ফিজিকাল ডিভাইস বা ভৌত উপাদানগুলো কাজ করে একদম ভৌত রকমে, র মানে কাঁচা লেভেলে। যেখানে ঘাম ঝরে, রক্ত পড়ে, ঠ্যাং ভেঙে হাড় বেরিয়ে যায়। সেইসব ভৌততা ছেড়ে তাদের কাজকে ভৌতিক মানে সাঙ্কেতিক তথ্যের আকারে নিয়ে আসে এই কন্ট্রোলারনিচয়। পরে ভালো করে বুঝব, ব্যাপারটা অনেকটা এরকম। আপনার মনে একটা আবেগ এল, এক লাইন কবিতা লিখতে চাইলেন, এটা একটা বিমূর্ত ব্যাপার, একটা চিন্তা। আপনি কিবোর্ডে টাইপ করলেন, সেখান থেকে সেটা গেল সরাসরি সিপিইউ-র এক্তিয়ারে, সিপিইউ এবার যে শব্দগুলো অক্ষরগুলো মিলে কবিতা, তাদের তথ্যটা পাঠাল বাসে, বাস থেকে যাবে স্ক্রিন বা কনসোল বা সামনে টাঙানো কাঁচের পর্দার তলে ফুটে ওঠার জন্যে। আপনার কাব্যচিন্তার মত এই তথ্যটাও বিমূর্ত, কিন্তু একটু অন্যরকমে, সিপিইউ তাদের কিছু সংখ্যা বানিয়ে দিয়েছে, যে সংখ্যাগুলোকে পড়ে তাদের পিছনে মূল অক্ষরগুলোকে চিনে নিতে পারবে আপনার ডিভাইসের কন্ট্রোলার। তার জন্যে প্রথমে বাস থেকে তথ্যটা গেল আপনার মেশিনের স্ক্রিন নামক ডিভাইসের কন্ট্রোলারের কাছে, সেই কন্ট্রোলার সেই বিমূর্ত সংখ্যাগুলোকে পড়ল, পড়ে বুঝল সিপিইউ কোন অক্ষরগুলোকে ফুটিয়ে তুলতে চাইছে স্ক্রিনে। কিন্তু কন্ট্রোলার এবার যা পাঠাবে তা আর বিমূর্ত নয়, একদম আছোলা কিছু বৈদ্যুতিক আদেশ, স্ক্রিন জুড়ে থাকা ছোট ছোট চিত্রকোষ বা পিকসেলের মধ্যে এই এই পিকসেলে এত এত মাত্রার বিদ্যুৎ পাঠাও, যাতে স্ক্রিন জুড়ে আপনার লেখা কবিতার লাইনটা ভেসে উঠতে পারে। সেই ভেসে ওঠাটা একদমই ভৌত, আপনার স্ক্রিনের ভৌত গঠনে কোনো গোলযোগ যদি থাকে, সেটা আপনার কবিতার লাইনকে ভেস্তে দেবেই, কবিতাটা যতই উৎকৃষ্ট হোক না কেন।

কন্ট্রোলারগুলোর সঙ্গে যুক্ত এই বাসের সঙ্গেই আবার লাগানো থাকে সিপিইউ এবং মেমরি। সিপিইউ তো লাগানো থাকতেই হবে, সেই কম্পিউটারের সকল কাজের কাজী, এইমাত্র এই কাব্যচর্চার উদাহরণটাকে ভাবুন, আর মেমরি লাগে সিপিইউর প্রতি মুহূর্তের প্রতিটি কাজের জন্যে। তাই এদের দুজনকেও বাসের সঙ্গে সংযুক্ত না-থাকলে কিছুতেই চলবেনা।

সিপিইউ বা সেন্ট্রাল প্রসেসিং ইউনিট হল কম্পিউটারের মূল সক্রিয় অংশ — তথ্য চটকানোর কেন্দ্রীয় সংস্থা। শূন্য নম্বর দিনের ১ নম্বর সেকশনে আমরা কম্পিউটারের পাঁচ রকম উপাদানের ভিতর যাকে শেষ আলোচনা করেছিলাম — সব কাজের নিয়ন্ত্রণ করাই যার কাজ, সেই কেন্দ্রীয় নিয়ন্ত্রণ সংস্থা এই সিপিইউ। এই সিপিইউ-র মধ্যেই ভরা থাকে কম্পিউটারের ঘড়ি। যারা শুধু গোটা গোটা অক্ষরে স্ক্রিনে তারিখ এবং সময় ফুটিয়েই তোলে না, কম্পিউটারের ভিতর চলমান প্রতিটি প্রোগ্রামের সময়কাঠামোকে নিয়ন্ত্রণ করে।

আমরা পরে দেখব প্রচুর প্রক্রিয়া চলে একটা গ্নু-লিনাক্স মেশিনে যারা সরাসরি এই ঘড়িগুলোর দিকে তাকিয়ে থাকে অপলক চোখে — ক্রন, অ্যাট, ইত্যাদি, ঘড়ি ঘড়ি তাদের দিল ধড়কানোই নিয়ম। এই সময়ানুবর্তিতার পাশাপাশি সিপিইউতে থাকে একজন একনিষ্ঠ একাউন্টান্ট। দিবারাত্রি সে শুধু হিশেবই করে, তার নামটা একটু অ্যান্টিক্লাইম্যাক্স যদিও, মানে বাংলায় পড়লে, তার নাম আলু। আমরা আগেই বলেছি। অ্যারিথমেটিক অ্যান্ড লজিকাল ইউনিট। শুধু যোগ বিয়োগ গুণ ভাগই করে না, লজিকাল হিশেবও করে। হ্যাঁ না না, ঠিক না ভুল, ইত্যাদি। আর থাকে নিয়ন্ত্রক সংস্থা। কন্ট্রোল এজেন্সি। প্রোগ্রামের মধ্যেকার নির্দেশগুলো বুঝে যে প্রোগ্রামগুলোকে চালায়। আর থাকে সার্কিট মেমরি থেকে তথ্য নিয়ে আসার। নির্দেশ আর তথ্য — এই দুটোকে ভরা থাকে মূল মেমরিতে মানে র‍্যাম বা রম-এ। প্রয়োজন অনুযায় সিপিইউ তাদের নিয়ে আসে।

বহিরাঙ্গিক উপাদান বা পেরিফেরাল ডিভাইসগুলোর এক একটা করে নিয়ন্ত্রক সংস্থা থাকে। এই কন্ট্রোলারগুলো তাদের নিজের নিজের অধীন উপাদান থেকে, ধরুন ফ্লপির কন্ট্রোলার ফ্লপি থেকে, মাউস কন্ট্রোলার মাউস থেকে তথ্যগুলো নিয়ে আসে বাস-এ। বা তথ্য বার করে নিয়ে যায়। কনসোল কন্ট্রোলার কনসোলে নিয়ে যায় ফুটিয়ে তোলার তথ্য এই বাস থেকেই। ইনপুট আর আউটপুট — তথ্য ঢোকানো বা বার-করার ডিভাইসের এই দু-রকম তথ্যেরই মধ্যস্থতা করে এই কন্ট্রোলাররা। কিবোর্ড বা মাউস জাতীয় ইনপুট ডিভাইস, বা প্রিন্টার, কনসোল জাতীয় আউটপুট ডিভাইস, বা হার্ড ডিস্ক জাতীয় তথ্যের ভাঁড়ার — এই সবগুলো ডিভাইসই সরাসরি কথা বলে তার কন্ট্রোলারের সঙ্গে, কন্ট্রোলার তার পর তথ্যটাকে নিয়ে যায় বাসে। সিপিইউ আর পেরিফেরাল ডিভাইস — এরা কাজ করে একই সঙ্গে — কম্পিউটার আর তার তার ব্যবহারকারীর পারস্পরিকতা বা ইন্টারফেসটাকে বানিয়ে তোলে। ব্যবহারকারী তার মেশিনের সঙ্গে যাতে একটা কথোপকথনে আসতে পারে। তথ্যের সঞ্চারপথটা হয় এরকম — ইনপুট ডিভাইস থেকে কন্ট্রোলার থেকে বাস হয়ে সিপিইউ, বা সিপিইউ থেকে বাস হয়ে কন্ট্রোলার হয়ে আউটপুট ডিভাইস। কখনো আবার সিপিইউ সরাসরি এই তথ্যকে সরাসরি নিজের কাছে না-এনে পাঠিয়ে দেয় মেমরিতে, যাতে পরে সে মেমরি থেকে পড়ে নিতে পারে।

১.১।। সিপিইউ এবং তার রেজিস্টার-কাঠামো

আধুনিক একটা সিপিইউ মানে একটা ছোট্ট সিলিকন চিপ। এই চিপের উপরে এঁকে বা দেগে বা ছেপে দেওয়া লক্ষ লক্ষ ট্র্যানজিস্টর এবং তাদের মধ্যেকার বৈদ্যুতিক সংযোগ — যার গোটাটা মিলিয়ে সিপিইউর মোট সার্কিট। আমরা চিপের ছবি, তার আকার আয়তন নিয়ে কথা বলেছি শূন্য নম্বর দিনেই। চিপের শরীরের ধার ঘেঁষে লাগানো থাকে একশো বা তারও বেশি পিন। এর কিছু পিন সংযোগ করে বাস-এর সঙ্গে, তথ্য যাতায়াত করে এই পিনগুলো বেয়ে।

সিপিইউ-র প্রাথমিক অংশগুলো

সময়-রাখার এবং নিয়ন্ত্রণের সার্কিট
আলু বা আঙ্কিক-যৌক্তিক হিশেবের সার্কিট
দ্রুতগতি রেজিস্টারের ভাঁড়ার

আর অন্য পিনগুলো কম্পিউটারের বিদ্যুৎ-লাইন থেকে বিদ্যুৎ বয়ে নিয়ে যায় চিপ অব্দি।

ভৌত গঠনের নিরিখে একটা চিপের শরীর গঠিত হয় অনেকগুলো স্তরে অনেকগুলো জটিল রাসায়নিক এবং বৈদ্যুতিক এবং বৈদ্যুতিন প্রযুক্তিকে মিলিয়ে। কিন্তু, কাজের ছকের ভিত্তিতে একটা সিপিইউ চিপের মূল অংশ সবসময়েই এই তিনটে —

(ক) সময়-রাখার এবং নিয়ন্ত্রণের সার্কিট — সেই সব সার্কিট যারা সময়কে খেয়াল করে, ধারণ করে, এবং করে চলে কন্ট্রোল বা নিয়ন্ত্রণের কাজ, মানে প্রোগ্রাম বা সফটওয়ারগুলোর মধ্যে লিপিবদ্ধ নির্দেশগুলোকে লাগু করে। এই টাইমিং এবং কন্ট্রোল সার্কিট হল কম্পিউটারের জরুরিতম অঙ্গ সিপিইউ-র আবার সবচেয়ে জরুরি অংশ। ধরুন শেষ অব্দি কম্পিউটারের কম্পিউটার অংশ কতটুকু? তার উত্তর হবে এই টাইমিং এবং কন্ট্রোল সার্কিট। দিন শূন্য থেকে ক্যালকুলেটর আর কম্পিউটারের পার্থক্যের আলোচনা মনে করুন। একটা সময়-রাখা এবং নিয়ন্ত্রণের সার্কিটের কাজের গড়ন অনেকটা এইরকম হতে পারে —

repeat — এর মানে, এইখান থেকে এর পরের অংশ বারবার করে চলবে সিপিইউ, হল্ট অব্দি।

fetch next instruction from memory — মেমরিতে তুলে রাখা প্রোগ্রামের পরের নির্দেশটা পড়ো।

decode instruction — নির্দেশ অনুযায়ী ঠিক কোন তথ্যকে কোন সার্কিট দিয়ে চটকাতে হবে সেটা স্থির করো।

fetch from memory additional data — মেমরি থেকে, যদি লাগে, আরো তথ্য নিয়ে এসো।

execute the instruction — সঠিক সার্কিটে সঠিক তথ্যটা পৌঁছে দাও, যাতে সে কাজ করতে পারে।

until halt — এর মানে এই অব্দি এসে বৃত্তটা সম্পূর্ণ হল, রিপিট থেকে যা শুরু — এটা একটা লুপ, পরে আসব আমরা।

শুধু এই ‘ফেচ-ডিকোড-একজিকিউট’ বা ‘আনো-বোঝো-পালন-করো’ জাতীয় নির্দেশ-বৃত্ত বা লুপই নয়, এই টাইমিং এবং কন্ট্রোল সার্কিটেরই আভ্যন্তরীন হল সেইসব সার্কিট যারা নির্দেশগুলোকে ডিকোড করে মানে বোঝে, আর সেই নির্দেশের মধ্যে আবদ্ধ থাকা ঠিকানাগুলোকে চিনতে পারে। নইলে দেখুন না, সে যদি নিতান্তই সময় রাখা আর নিয়ন্ত্রণের বাইরে আর কিছু না-পারে, তাহলে ওই বৃত্তটার তিন-নম্বর লাইনে, মানে ‘ডিকোড’-এ এসে তো আটকে যাবে — ভাই বোঝার দায়িত্ব তো আমার নয় — আমার দায়িত্ব তো করা? আর বোঝার মধ্যেই রয়ে যাচ্ছে কিছু ঠিকানা চিনে নেওয়া, কারণ, এর পরের লাইনেই আসছে, প্রয়োজন মোতাবেক আরো তথ্য নিয়ে এসো। নিয়ে এসো তো বললাম, আনবে কোত্থেকে? আর আধুনিক একটা চিপ, সাইজ তো আগেই দেখলেন, ভারি চিকণ জিনিষ, হনুমান নয় যে গন্ধমাদন উপড়ে আনবে বিশল্যকরণীর সঠিক বাস-রুট পায়নি বলে। এই যে লুপ নিয়ে একটু কথা আমরা বলে নিলাম, এই প্রসঙ্গটা বারবার ফিরে ফিরে আসবে আমাদের গোটা পাঠমালাটা জুড়ে।

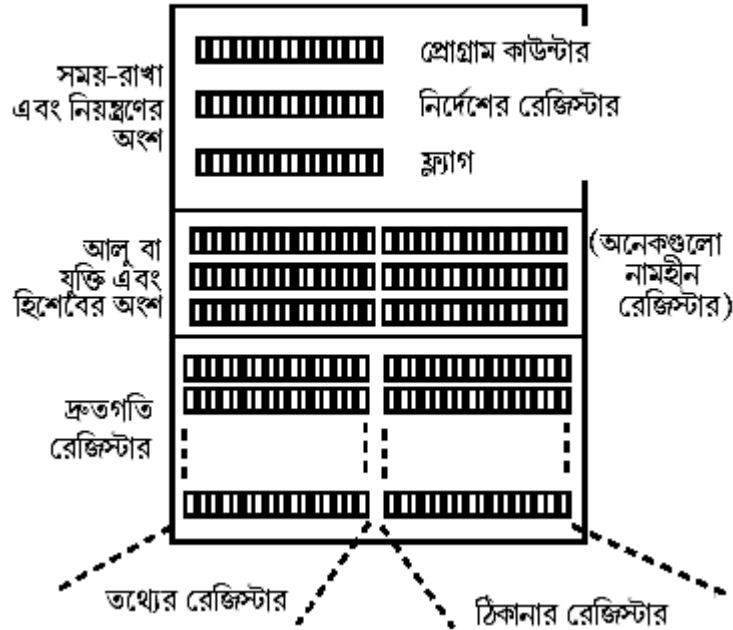
(খ) আলু বা আঙ্কিক-যৌক্তিক হিশেবের সার্কিট — আলু বা অ্যারিথমেটিক লজিক ইউনিটের মধ্যে থাকে সেইসব সার্কিট যারা আঙ্কিক তথ্যকে ইধার-কা-মাল-উধার, মানে ম্যানিপুলেট করে, নাড়াচাড়া করে। যোগ বা গুণ করার জন্যে নির্দিষ্ট সার্কিট আছে। অনেকসময়েই এই সার্কিটগুলো শুধু এক রকমের না, একাধিক রকমের হয়। কিছু সার্কিট হয় যারা পূর্ণসংখ্যা মানে ইন্টিজারদের নিয়ে নাড়াচাড়া করে। কিছু সার্কিট করে রিয়াল নাম্বার বা বাস্তব সংখ্যাকে। কিছু সার্কিট আছে তুলনা করার জন্যে। একটা নির্দিষ্ট ভ্যারিয়েবল একটা নির্দিষ্ট মানের চেয়ে ছোট না বড় না সমান — প্রোগ্রামগুলোর এটা বোঝার দরকার পড়ে হরহামেশাই। সেই কাজটা করে দেন এই আলুবাবু। ডেটা বা তথ্যের একটা নির্দিষ্ট আকৃতি বা বিট প্যাটার্ন আছে কিনা সেটারও বিচার করেন। দরকার পড়লে সেই আকৃতি বা বিট-প্যাটার্নটা বদলাতেও পারে আলুবাবুর লজিক সার্কিটরা।

(গ) দ্রুতগতি রেজিস্টারের ভাঁড়ার — কম্পিউটার তার প্রয়োজনের বেশিরভাগ তথ্যটাই রাখে মেমরিতে। কিন্তু সামান্য কিছু তথ্য রাখার ব্যবস্থা থাকে সিপিইউ-র ভিতরেই — রেজিস্টারে। মূল মেমরিটা যেখানে কোটি কোটি তথ্য একক ধরে রাখতে পারে, সেখানে এই রেজিস্টার পারে ধরুন ষোলটা তথ্য একক। সাত নম্বর দিনে, ভারচুয়াল মেমরির প্রসঙ্গে আমরা এই বিভিন্ন ধরনের মেমরির একটা তুলনামূলক আলোচনা করব। একটা সিপিইউ রেজিস্টার রাখতে পারে কম্পিউটারের একটা ওয়ার্ড বা শব্দর যা মাপ ঠিক সেই পরিমাণ তথ্য। ওয়ার্ড এর সংজ্ঞা দেখুন শূন্য নম্বর দিনের শেষে তথ্য পরিমাপের এককে। আর একবার মনে

> করিয়ে দিই, একটা বিট মানে নিছক একটা ০ বা ১, একটা বাইট মানে আটটা বিট-এর একটা সমাহার, একটা ওয়ার্ড কম্পিউটার থেকে কম্পিউটারে বদলায়, ষোল বিট হতে পারে, মানে দুই-বাইট বা বত্রিশ বিট হতে পারে, মানে চার বাইট, বা এমনকি তার বেশিও হতে পারে। বেশিরভাগ আধুনিক সিপিইউ-র বেলায় এই রেজিস্টারের মাপ হল বত্রিশ বিট করে। আলুর সার্কিট এমনভাবেই তৈরি এবং সমাবদ্ধ যে আলুর কাছে যে তথ্য যায়, বা আলু থেকে যে তথ্য বেরোয়, তার অনেকটা বা পুরোটাই — গোটা ইনপুট আউটপুটটাই ঘটে সিপিইউ রেজিস্টার মারফত। তথ্যটা মেমরি থেকে তুলে এনে চকিত চোখের চঞ্চল চাহনির মত একটা ছোট্ট মুহূর্ত ধরা থাকে সিপিইউর রেজিস্টারে, তারপরেই মালটা চলে যায় আলুর দখলে। আর হিশেবের কাজ শেষ হয়ে গেলে ফেরত এনে তুলে রাখা হয় মূল মেমরির আলমারিতে। সিপিইউর কিছু রেজিস্টার থাকে এই তথ্য মানে বিট-মানগুলোকে ধরে রাখার জন্যে, আর কিছু রেজিস্টার থাকে সিপিইউ কাজ করে চলাকালীন মেমরির কোন কোন ঠিকানায় কী কী করতে হবে সেই হিশেব নিকেশ করার জন্যে। রেজিস্টারগুলো কী ভাবে ব্যবহার হবে তার ভিত্তিতে সিপিইউগুলোর গঠনের তফাত হয়, কিন্তু, মোটামুটিভাবে, বেশিরভাগ সিপিইউতেই আটটা বা তার বেশি রেজিস্টার থাকে তথ্য রাখার, আর আরো আটটা থাকে ওই ঠিকানার জন্যে।

এই তথ্য আর ঠিকানার রেজিস্টার ছাড়াও আরো রেজিস্টার থাকতে পারে একটা সিপিইউতে — আলু আঙ্কিক হিশেব করে চলাকালীন ছোট ছোট 'হাতে-রইল-পেন্সিল'-এর জন্যে। আবার সময়-রাখার এবং নিয়ন্ত্রণের সার্কিটেও থাকে কিছু রেজিস্টার — নিয়ন্ত্রণ সংক্রান্ত তথ্য রাখার জন্যে। সঙ্গের ছবিতে দেখুন টাইমিং এবং কন্ট্রোল ইউনিট বা সময়-রাখা এবং নিয়ন্ত্রণের অংশে তিন রকম রেজিস্টার হয় — প্রোগ্রাম কাউন্টার বা প্রোগ্রামের হিশেবরক্ষক, ইন্সট্রাকশন বা নির্দেশের রেজিস্টার এবং ফ্ল্যাগ। এই প্রোগ্রাম কাউন্টার ধরে রাখে মেমরির সেইসব এলাকার ঠিকানা যাতে ঠিক এরপরেই যে নির্দেশটা পালন করতে হবে সেটা যেখানে লেখা আছে। নির্দেশ বা ইন্সট্রাকশন রেজিস্টার ধরে রাখে এই মুহূর্তে যে নির্দেশটা পালন করা হচ্ছে তার বিট সমাহার। এছাড়াও বেশিরভাগ সিপিইউতেই থাকে

সিপিইউ রেজিস্টারগুলোর আলাদা আলাদা কাজ

সময়-রাখা এবং নিয়ন্ত্রণের অংশ
প্রোগ্রাম কাউন্টার
নির্দেশের রেজিস্টার
ফ্ল্যাগ
আলু বা যুক্তি এবং হিশেবের অংশ
(অনেকগুলো নামহীন রেজিস্টার)
দ্রুতগতি রেজিস্টার
তথ্যের রেজিস্টার
ঠিকানার রেজিস্টার

একটা ফ্লাগ রেজিস্টার। মানে ধরুন অবস্থান অ্যাকাউন্টান্ট। যে নির্দেশগুলো পালন করা হচ্ছে তাদের অবস্থাটা লিখে রাখে এই ফ্লাগ। একটা বিট থাকে যেটা দেখায় যে ওই পালনমান নির্দেশটা সিপিইউতে এসেছে কোনো সাধারণ প্রোগ্রাম থেকে না সরাসরি অপারেটিং সিস্টেম থেকে।

এই অস (OS — **O**perating-**S**ystem) বা অপারেটিং সিস্টেম হল কম্পিউটারের ভিতরে সকল মাস্তানের বাবা মাস্তান, অন্য সকল প্রোগ্রামাস্তানেরা যেখান থেকে প্রবাহিত হয়, সেই নেতার নেতা, মানে ধরুন, রাজনীতির মুখ্যমন্ত্রী বলা যায়। তার অনেক কিছু করার রাইট থাকে যা অন্য প্রোগ্রামের কাছে রং, ওখানে কোনো রংবাজি চলবে না কাকা।

আমাদের পরে এটা খুব ভালো করে বুঝতে হবে, বিশেষত একটা গ্নু-লিনাক্স সিস্টেমের সাপেক্ষে। এখানে যে কথাগুলো আমরা বলছি, সেগুলো সবটাই হল ভিতখোঁড়া, পরে এর উপরে আমরা গ্নু-লিনাক্স অপারেটিং সিস্টেমের কাঠামোটা গজিয়ে উঠতে দেখব। আর খুব বেশি করে এই সিপিইউ রেজিস্টারের আর ফ্ল্যাগের ধারণাগুলো যেখানে জরুরি হয়ে ওঠে, সেটা হল মূলত গ্নু-লিনাক্স অ্যাজ এ প্রোগ্রামিং এনভায়রনমেন্ট — যা তথাগতর ভবিষ্যদ্বানী অনুযায়ী ৩৫ বা ৩৬ নম্বর দিনের আগে আসার চান্স নেই। তার মানেই চান্স নেই। কারণ এই বইটা তার চেয়ে অনেক ছোট হবে, প্রোগ্রামিং অর্থে একটু ব্যাশ স্ক্রিপ্টিং আসবে, তাও এই পাঠমালার দশ নম্বর মানে শেষ দিনে গিয়ে।

ফ্লাগ ব্যাপারটাও অসম্ভব জরুরি হয়ে ওঠে প্রোগ্রামিং বোঝার বেলাতেই। ইন ফ্যাক্ট এই সেকশনের ছবির ছকগুলো এবং বক্তব্যেরও বেশিরভাগটাই নেওয়া একটা সিপ্লাসপ্লাস প্রোগ্রামিং-এর বই থেকে, যার নামটা আমার মনে নেই, ডাউনলোড করা ফাইলটাও ভুল বশত উড়ে গেছে। নেটে সিপ্লাসপ্লাস দিয়ে সার্চ মারলে পেয়ে যাওয়ার কথা, নয়ত কারোর লাগলে আমার মূল নোটসটা দিতে পারি। সেখান থেকে নেওয়া নোটসটাই এখানে কাজে লাগছে। না ভাই, ঝাড়া বলবেন না, কানাকে কানা বলতে নেই, আগেই বলেছি, অন্নু মালিক আমাদের রোল মডেল। আপনাদের কাছে খিস্তি খাওয়ার ভয়ে দেখুন, প্রতিটি দিনের শেষে লেখা ‘রচনা ও সংকলন’, শুধু ‘রচনা’ নয় বস।

যাই হোক, যে ফ্লাগ নামক রেজিস্টারগুলোর কথা বলছিলাম, এদের উপর পুরো স্বৈরতান্ত্রিক নিয়ন্ত্রণ চলে অপারেটিং সিস্টেমের। এই অপারেটিং সিস্টেমের অনেক বাড়তি অধিকার থাকে, শুধু এই ফ্লাগদের উপর নয়, বহিরাঙ্গিক বা পেরিফেরাল উপাদানগুলোর কন্ট্রোলারদের কাজকর্মের উপরও, যা অন্য সব আমপ্রোগ্রামের থাকেনা। সাধারণ প্রয়োগমূলক বা অ্যাপ্লিকেশন প্রোগ্রামগুলোর যখন এদের কোনোটাকে ব্যবহার করার প্রয়োজন পড়ে, তারা আবেদন জানায় কারনেলের বা অপারেটিং সিস্টেমের কাছে। কারনেল হল সকল কাজের কাণ্ডারী, সকল কোষের ভাণ্ডারী। সে ঠিক করে দেয় কখন কোন প্রোগ্রাম কতটুকু অব্দি এদের ব্যবহার করতে পারবে। পরের দিন, মানে, এই পাঠমালার দুই নম্বর দিনে এটা নিয়ে আর একটু আলোচনা করব আমরা, তারপর কম্পিউটার বিবর্তনের ইতিহাসের সঙ্গে মিলিয়ে তাদের বুঝব চার নম্বর দিনে গিয়ে। কারনেলের এই চূড়ান্ত নিয়ন্ত্রণটাই ঘোষিত থাকে ফ্লাগ রেজিস্টারদের একটা বিটে। অন্য বিটগুলোর গায়ে লেখা হয় আলুবাবুর বিভিন্ন তুলনামূলক কাজকর্মের ফসলাবলী। ধরুন একটা ভ্যারিয়েবল যদি একটা সংখ্যার সমান হয় — সেটা লেখা থাকবে একটা বিটে, অন্য একটায় লেখা থাকবে যদি সেটা বড় হয় — ইত্যাদি। মানে একটা বিট হল নিয়ন্ত্রণের বিট, অন্যগুলো হল কাজের ফলাফলের বিট।

আসলে একটা অন থাকা কম্পিউটার মানে তো কিছু চলমান প্রোগ্রামের সমষ্টি। এবং কিছু ভৌত উপাদানের শরীরে প্রতিমুহূর্তে লেখা হতে থাকা আঁকা হতে থাকা এই প্রোগ্রামগুলোর সঞ্চারপথ। আর একটা প্রোগ্রাম মানেই কিছু পরপর লিখে রাখা নির্দেশের তালিকা। যা মেমরিতে তুলে ফেলা হয়েছে এবং সিপিইউ পরপর তাকে পালন করে যাচ্ছে। আর এই করতে গিয়ে যে যে পেরিফেরাল ডিভাইসকে তার যে যে ভাবে যতক্ষণ ধরে কাজে লাগানো দরকার সেই সেই ভাবে তাদের সক্রিয় করে তোলা হচ্ছে। ঠিক এই কাজগুলোকে পরপর চালিয়ে যাওয়াই সিপিইউর রেজিস্টারগুলোর কাজ। আপাতত এটা যথেষ্ট হয়েছে — এবার দু-একটা কথা বলে নেওয়া যাক সিপিইউর সঙ্গে মেমরির সম্পর্ক নিয়ে। আমরা এতবার করে বলছি, মেমরির ঠিকানা, মেমরি থেকে তথ্য তোলা — এর ঠিক মানেটা কী। সিপিইউ ঠিক কী ভাবে এটা করে।

১.২।। মেমরি এবং তথ্য

আমরা আগেই বলেছি, কম্পিউটারের দু-রকম মেমরি — রম (ROM — **R**ead-**O**nly-**M**emory) আর র‍্যাম (RAM — **R**andom-**A**ccess-**M**emory)। রম মানে যেখান থেকে সিপিইউ শুধু পড়ে। আর র‍্যাম মানে যেখানে, যেমন দরকার, পড়ে বা লেখে। মেশিনের প্রাথমিকতম রম আর র‍্যাম থাকে মাদারবোর্ডের চিপে।

আর এর বাইরে আর এক রকম মেমরি হল তথ্যের ভাঁড়ার বা স্টোরেজ, হার্ড ডিস্ক, সিডি, ফ্লপি ডিস্ক, জিপ ড্রাইভ জাতীয়। যাকে সচরাচর আমরা যখন কম্পিউটারের মেমরি বলে উল্লেখ করি তার মধ্যে ধরিনা। র‍্যাম-এর নাম র‍্যাম হয়েছে মূলত এই ধরনের ভাঁড়ার বা স্টোরেজ ডিভাইস থেকে একে আলাদা করতে গিয়ে। এখান থেকে যে কোনো ভাবে তথ্য পড়া যায় — যে কোনো বিটের পরে যে কোনো বিট — এই অর্থে এটা র‍্যানডমলি অ্যাকসেসিবল।

রম সচরাচর একটা কম্পিউটারের অপারেটিং সিস্টেমের প্রাথমিকতম অংশগুলোকে ধরে রাখে। অপারেটিং সিস্টেম নিয়ে একেবারে কিছু না-বলে এদের নিয়ে কথা বলা শক্ত। বারবার খুব অসুবিধা হচ্ছে কথা বলতে গিয়ে, কিন্তু কিছু করার নেই, ঠিক এরের পরের ২ নম্বর সেকশন থেকেই আমরা আসব অপারেটিং সিস্টেমের আলোচনায়। দুই থেকে পাঁচ নম্বর দিনের গোড়া অব্দি চলবে এই অপারেটিং সিস্টেম কী করে গজিয়ে উঠল তার ইতিহাস। সেই ইতিহাস যৌথ ভাবে কম্পিউটারের হার্ডওয়ার এবং সফটওয়ারের ইতিহাস। তারপর গোটা পাঠমালা জুড়ে বারবার আসতেই থাকব একটা বিশেষ রকম অপারেটিং সিস্টেম — গ্নু-লিনাক্স অপারেটিং সিস্টেমের আলোচনায়।

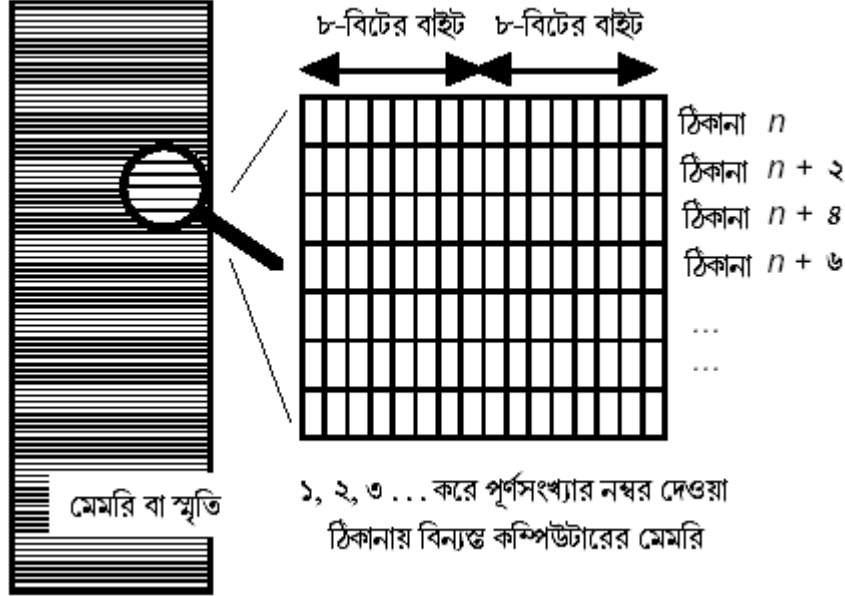
অপারেটিং সিস্টেমটাও একটা প্রোগ্রাম। সব প্রোগ্রামের কেন্দ্রীয় প্রোগ্রাম। এই অপারেটিং সিস্টেমের প্রায় গোটাটাই লেখা থাকে হার্ড ডিস্কে। কম্পিউটার অন করার পর থেকে কাজ করার অবস্থায় আসা অব্দি গোটা প্রক্রিয়াটার নাম বুট। এই বুট করাকালীন কী ভাবে রম চিপে লেখা থাকা নির্দেশ মেনে হার্ড-ডিস্কে লিখে রাখা অপারেটিং সিস্টেমকে সক্রিয় এবং জীবন্ত করে তোলে সিপিইউ, প্রাণ-প্রতিষ্ঠা করে সিলিকন-তার-চাকতি-প্লাস্টিক-স্ক্রু ইত্যাদি দিয়ে তৈরি ওই দামী এবং জটিল খেলনায় — সেই আলোচনা করব আমরা পাঁচ নম্বর দিনে।

শূন্য নম্বর দিনে আমরা উল্লেখ করেছি, একটা কম্পিউটারে স্বভাবত র‌্যামের মোট পরিমাণের তুলনায় রমের পরিমাণ যৎসামান্য। মেমরির সিংহতমভাগই এই র‌্যাম। র‌্যাম তার চিপের পর চিপ জুড়ে ধরে রাখে অপারেটিং সিস্টেমের মোট কোড এবং তথ্য। কোড বলতে এখন থেকে আমরা বুঝব প্রোগ্রামকে, প্রোগ্রাম মানে তো পরপর লিপিবদ্ধ নির্দেশ — সেই নির্দেশগুলোর ক্রমানুসারী তালিকাটাকে এখন থেকে আমরা কোড বলে ডাকব। সেটাই চালু নাম। শুধু এই অপারেটিং সিস্টেমের কোড এবং সেই কোডগুলো চালানোর জন্যে প্রয়োজনীয় তথ্যই নয়, অপারেটিং সিস্টেম এরপর যে প্রোগ্রামগুলোকে চালাবে সেই প্রোগ্রামগুলোর কোড এবং তথ্যও একই ভাবে তুলে নিতে হয় র‌্যামকে। মেমরির পরিমাণের একক, আমরা বলেছি, বিট বাইট এবং ওয়ার্ড। বিট বা বাইট তো সব কম্পিউটারেই এক, কিন্তু ওয়ার্ডের মাপ কম্পিউটার থেকে কম্পিউটারে বদলায়। দুই বাইট থেকে চার বা আট বাইট, ১৬ বা ৩২ বা ৬৪ বিট, বা হয়ত আরো বেশি আসবে আগামী দিনে। মেমরির সাইজ উল্লেখ করা হয় বাইট দিয়ে। বা ১০২৪ গুণ ১০২৪ মানে ১০৪৮৫৭৬ বাইট বা এক মেগাবাইট দিয়ে। বিট দিয়ে উল্লেখ করা মানে ধরুন ১২৮ এমবি-র একটা র‌্যামের সাইজ দাঁড়াবে ১০৭৩৭৪১৮২৪ বিট — দেখে নিয়ে আর না-তাকিয়ে একবারে বলুন তো আপনার মেশিনের র‌্যামের সাইজ কত বিট?

এই র‌্যাম বলুন, রম বলুন, স্টোরেজ ডিভাইস বলুন, সবাই শেষ অব্দি কাজ করে বিট দিয়ে। একটা বিট মানে একটা ছোট্ট খোপ — যাতে থাকতে পারে হয় ০ নয় ১। তার মানে খোপটার দুটো ভৌত অবস্থা থাকতে পারে, একটা ভৌত অবস্থা হাজির করবে শূন্য, অন্য ভৌত অবস্থা হাজির করবে এক। কী ভাবে এই অবস্থাটা তৈরি হবে তার অনেকরকম প্রকৌশল আছে। চার নম্বর দিনে আমরা যখন প্রাচীনতম কম্পিউটারগুলোর আলোচনা করব তাতে আসবে ভালভসেট দিয়ে বানানো মেশিনের কথা, যেখানে একটা পারদের টিউবের ভিতর একটা বিদ্যুৎ-উপস্থিতি মানে এক, আর তাই, না-থাকা মানে শূন্য। ভাঁড়ার তখন রাখা হত মূলত মেশিনের বাইরে, কাগজে বা কার্ডে, এই কার্ডে আবার একটা ফুটো থাকা মানে শূন্য, ফুটো না-থাকা মানে এক। ক্যাসেট টেপের ফিতেয় রাখা মানেও তাই — কোনো একটা চৌম্বক উপস্থিতি মানে ১, অনুপস্থিতি মানে ০। এর পরে এসেছিল চৌম্বক অক্সাইডের ছোট ছোট লুপ — যার নাম ছিল কোর, যার উত্তর-দক্ষিণ মেরু-সংস্থান দিয়ে হাজির করা হত ০/১। এই নামটা আজো চলছে। ইউনিক্স জগতে, ওপন-সোর্স-কোড ইউনিক্স মানে লিনাক্স জগতেও, এখনো র‌্যামকে অনেকসময়ই ডাকা হয় কোর বা কোর মেমরি বলে। যদিও এখন এই প্রকৌশল আমূল বদলে গেছে। তবে হার্ড ডিস্কে যেভাবে তথ্য রাখা হয় সেটাও এক ধরনের একটা চৌম্বক পদ্ধতি।

প্রকৌশলের সবচেয়ে নাটকীয় বদলটা এসেছিল ইন্টিগ্রেটেড সার্কিট থেকে। শূন্য নম্বর দিনে আমরা ছবি দিয়ে দেখিয়েছি — এখনকার র‌্যাম তৈরি হয় এই আইসি বা ইন্টিগ্রেডেট সার্কিট চিপ দিয়ে। এর ভিতর খুব প্রাথমিক একটা সার্কিটকে বলে ফ্লিপ-ফ্লপ সার্কিট। তার দিয়ে জোড়া চারটে ট্রানজিস্টর মিলিয়ে সার্কিটটা — একটা একক, এটা অন থাকা মানে ১, আর অফ থাকা মানে ০। অর্থাৎ এই এককটা একটা এক বিট তথ্যকে ধরে রাখতে পারে। আর একটু বড় সাইজের সার্কিট, আটটা ফ্লিপ-ফ্লপের, রাখতে পারে এক বাইট তথ্য। এরকম অজস্র সার্কিট মিলিয়ে

মিলিয়ে তৈরি হয় র‍্যাম চিপ যা লক্ষ লক্ষ বাইট তথ্যকে ধরে রাখতে পারে। এরকম বেশ কয়েকটা চিপ মিলিয়ে তৈরি হয় একটা কম্পিউটারের র‍্যাম। শূন্য নম্বর দিনে র‍্যামের ছবিটা মনে করুন।

এবারে এই র‍্যামের কাজ করার ছকের ছবিতে দেখুন — একটা র‍্যামের আভ্যন্তরীন কাঠামো। যেখানে ওয়ার্ড ধরেছি আমরা ষোল বিটের মানে দুই বাইটের। এবার ছবির সঙ্গে মিলিয়ে ভাবুন, একটা র‍্যামকে ভাবা যেতে পারে পরপর ওয়ার্ডের পর ওয়ার্ডের একটা একটা একমাত্রিক সজ্জা। সজ্জাটার সঙ্গে মিলিয়ে প্রত্যেকটা ওয়ার্ডের একটা অবস্থান আছে — মানে একটা সূচক সংখ্যা বা ইনডেক্স। এক দুই তিন করে।

এই সূচক বা ইন্ডেক্সটাই হল র‍্যামের মধ্যের প্রতিটি ওয়ার্ডের আলাদা আলাদা ঠিকানা। এবং একটা ওয়ার্ডের মধ্যে আছে দুটো করে বাইট। বেশিরভাগ কম্পিউটারেই ব্যবস্থা থেকে আলাদা ভাবে একটা একক বাইটকেও পড়ে ফেলার। তার মানে প্রতিটি ওয়ার্ড বাড়ছে দুই বাইট করে। ২, ৪, ৬ — এই ভাবে। যদি সেই কম্পিউটারে ওয়ার্ড হয় চার বাইটের তখন এটা এগোবে ৪, ৮, ১৬ — এই ভাবে।

এই র‍্যামের পরে আসে ক্যাশে-র (Cache) প্রসঙ্গ। ক্যাশের উল্লেখ আমরা শূন্য নম্বর দিনেই করেছিলাম। কম্পিউটারের পাঁচ রকম স্মৃতির কথা বলতে গিয়ে। ক্যাশে শব্দটার ব্যুৎপত্তিগত অর্থ হল গোপন ভাণ্ডার। ক্যাশে মেমরিও ঠিক সেইরকমই। সাধারণ প্রোগ্রামারকে এর হদিশ রাখতে হয়না, আর প্রোগ্রাম না-বানিয়ে সে শুধু যদি প্রোগ্রাম ব্যবহারকারী হয় তাহলে তো জানার সম্ভাবনা আরো কম। ক্যাশে মেমরি থাকে কম্পিউটারের হার্ডওয়ারে, রাখা হয় শুধু দ্রুতগতিতে সিপিইউকে কাজ করানোর তাগিদে। পরে আমাদের বারবার আসতে হবে এগুলোর কথায়। তখন আমরা আরো জানব। ভারচুয়াল মেমরি বা ভৌতিক স্মৃতি — যা একটা গ্নু-লিনাক্স সিস্টেমের সবচেয়ে জোরের জায়গাগুলোর একটা, সেই ভারচুয়াল মেমরি, এবং তার ভৌত আকার মানে হার্ডডিস্কে লেখা সোয়াপ ফাইল বোঝার সময় আট নম্বর দিনে সবচেয়ে বিশদভাবে আসব আমরা এই আলোচনায়।

শুধু এবার মেমরি নিয়ে আমাদের এই আলোচনা-টাকে মিলিয়ে নিন শূন্য নম্বর দিনের একদম শেষে তথ্য তৈরির উপায় হিশেবে অ্যাস্কি কোডের আলোচনার সঙ্গে। সঙ্গের ছবির নিরিখে। আমরা আগেই বলেছি, কম্পিউটার তার ব্যবহার্য সমস্ত তথ্যকেই নাড়াচাড়া করে এবং চেনে বিট-সমাহার বা বিট-প্যাটার্ন দিয়ে। একটা বাইটে থাকে আটটা বিট। আটটা বিট মানে, আমরা দেখিয়েছি ২৫৬-টা আলাদা আলাদা বিট-সমাহার। নিজেই নিজেকে পরখ করে নিন, আপনি মনে করতে পারছেন কি, কোথা থেকে গজালো এই ২৫৬ সংখ্যাটা? জিএলটি ইশকুলে যাদের উপর অত্যাচার করার সুযোগ আমার হয়েছে, তাদের সবাইকে এই ধরনের প্রশ্ন মুহুর্মুহু এই ধরনের প্রশ্ন করে চমকে দিতে খুব ভালোবাসি আমি। এখানে আপনি নিজেই নিজেকে করুন। যা আলোচনা হয়েছে এখনো তাতে এটা আপনার মনে করতে পারা উচিত। একটা ইঙ্গিত দিই — '২' সংখ্যাটাকে '৮' সূচকে তুলে, মানে '$২^{৮}$' হল ২৫৬। এবার কি বুঝতে

পারছেন? একবার হিশেব করে দেখুন তো, বিটের সংখ্যা এক এক করে বাড়িয়ে, কটা করে আলাদা আলাদা সমাহার মানে সংখ্যা রাখা যাচ্ছে। আর এই সংখ্যাগুলো দিয়েই তো আমাদের মেশিনকে বোঝাতে হয় আর যা যা বোঝাতে চাই তার সমস্ত কিছু।

এই ২৫৬-টা সম্ভাব্য বিট-সমাহারের কয়েকটা আমরা এখানে দেখালাম। এক বাইট মেমরি-ভূমিতে যাদের রপ্তানি করা যায় সেরকম কিছু চিহ্ন এবং অক্ষর।

সম্ভাব্য ২৫৬-টা বিট-সমাহারের কয়েকটা

অপারেটিং সিস্টেমের আলোচনা শুরুর আগে সিপিইউ এবং মেমরি কিছু কথা আমরা বলে নিতে চেয়েছিলাম। সেই এক নম্বর দিনের এক নম্বর সেকশন এখানে শেষ। এবার আমরা যাব অপারেটিং সিস্টেম কাকে বলে তা নিয়ে একদম গোড়ার কিছু কথায়। যেখান থেকে আমরা অপারেটিং সিস্টেমের আলোচনা শুরু করতে পারব — যা এখন বেশ কয়েকদিন ধরে চলবে। অপারেটিং সিস্টেমের ইতিহাসের আলোচনা ধরে আমরা ইউনিক্স-এ পৌঁছব, তারপর ইউনিক্স থেকে মিনিক্স থেকে ওপন-সোর্স ইউনিক্স মানে গ্নু-লিনাক্স। যেখানে আমরা শেষ অব্দি পৌঁছতে চাই। আমাদের এই জিএলটি ইশকুলের পাঠমালায়। এই অব্দি আমরা হার্ডওয়ার নিয়ে কম্পিউটার সিস্টেম নিয়ে এমন বহু কথা বলেছি এবং আরো বলব যেগুলো সচরাচর কম্পিউটার ব্যবহারকারী হিশেবে আমাদের না-জানলেও চলে।

কিন্তু, প্রথমত আমরা অনেকেই যেখানে কাজ করে অভ্যস্ত সেই উইনডোজ পরিমণ্ডলের সঙ্গে গ্নু-লিনাক্স পরিমণ্ডলের একটা বড় পার্থক্য এই যে এখানে, গ্নু-লিনাক্সে যদি কেউ সিস্টেমকে তার গভীর অব্দি বুঝতে এবং ব্যবহার করতে চায় তার সেই সুযোগ আছে — যা উইনডোজ-এ নেই, গ্নু-লিনাক্স একটা কমিউনিটি, একটা আন্দোলন, উইনডোজ তা নয়। মাইক্রোসফট নামক একটা বহুজাতিক প্রতিষ্ঠানের ব্যবসার নাম উইনডোজ। তাই, তার ব্যবসার স্বার্থেই সে আপনাকে জানতে দেবে না, তার প্রোগ্রামগুলো একদম ভৌত স্তরে ঠিক কী ভাবে কাজ করে। পরে চার এবং পাঁচে এটা আমরা খুঁটিয়ে বুঝব। আর সেই জানাটা জানতে দেবেনা বলেই অপারেটিং সিস্টেমকে খুব তলিয়ে বোঝার চেষ্টা করাটাই উইনডোজ প্রতিবেশে একটা বোকামি, একটা জায়গায় গিয়ে সেটা আটকে যেতে বাধ্য। কিন্তু গ্নু-লিনাক্সে তা নয়, সবটাই খোলা প্রকাশিত মুক্ত, যতদূর আপনি জানতে চান, ততদূরই জানতে পাবেন। তাই যে জানাটা উইনডোজ প্রতিবেশে সম্পূর্ণ অকেজো সেটারই এখানে প্রচুর ব্যবহার আছে।

দুই, ঠিক ভাবে গ্নু-লিনাক্স বোঝাটাও উইনডোজ-এর চেয়ে অনেক বেশি রকম মেশিনমন্যতা দাবি করে। এর চেয়ে অন্য ভাবেও, মানেও ঠিক উইনডোজ-এর মত করেই, উইনডোজের চেয়ে আরো অনেকটা উন্নত এবং দক্ষ এবং নিরাপদ অন্যরকম একটা উইনডোজ হিশেবেই, গ্নু-লিনাক্সকেও, ব্যবহার করাই যায়, তাতে আর অতো মেশিন বোঝার দরকার পড়েনা। কিন্তু সে তো অনেকটা এমনি এমনি খাওয়া, নাতি বয়স থেকে দাদু বয়স অব্দি আপনি এমনি এমনিই খান, তার জন্যে এত লেখাপড়ার ছাই দরকার কিসে? এই বুড়ো বয়সে গ্নু-লিনাক্স শিখতে এসে আমার মনে হয়েছিল — সত্যিই — ঘরের কাছে আরশিনগরে এই যাদুকর পড়শির কথা — আর একটু আগে জানিনি কেন?

আর একটু ভালো করে শিখতে পারতাম। জীবন কাজ সবকিছুই আর একটু অন্যভাবে সমাহৃত করার চেষ্টা করা যেত।

কেউ যদি সেই যাদুটায় পৌঁছনোর চেষ্টা করেন, আমার চেয়ে অনেক তরুণ, এই লাগ-এর বেশির ভাগ ছেলে যেমন, অনেক জীবন্ত, এদের মতোই আরো অন্য নতুন কেউ, তাদের যদি একটু কাজ এগিয়ে দেওয়া যায়। কারণ, এটা করতে তো আমারও অসম্ভব পরিশ্রম হচ্ছে, কলেজের কলিগরা অনেকেই বলছে, করে তোমার লাভ কী, তোমার লাইনও তো না, পয়সাও পাবেনা, তাহলে? নিজেরও মনে হয় মাঝে মাঝে। গত কাল আমার পা ভাঙল, মধ্যমগ্রাম প্ল্যাটফর্মের গায়েই লাইটপোস্ট ধরে বসে পড়তে পড়তেই মনে এল, যদি ভেঙে থাকে, প্লাস্টার করতে করতে কলেজ যেতে যেতে অন্তত দিন সাতেক, তার মধ্যে এক দুই নেমে যাবে। সকালে এক্সরে করে ডাক্তার দেখিয়ে এলাম, সত্যিই ভেঙেছে — আজ এক নেমে যাচ্ছে। এখন বিকেল। পরের অংশটার প্রথম ড্রাফ্ট লেখা আছে। রাতের মধ্যে নেমে যাবে। এই তাগিদটা সেই যাদুর। একটা কমিউনিটি। একটা স্বপ্ন। তার নাম গ্নু-লিনাক্স। আমিও তার অংশ — আপনি হবেন কিনা, আপনি ঠিক করবেন।

২।। অ্যাপ্লিকেশন, সফটওয়ার, অপারেটিং সিস্টেম — সবই প্রোগ্রাম

এতক্ষণ অব্দি হার্ডওয়ার যে উপাদানগুলো নিয়ে আমরা আলোচনা করলাম — এদের মেলালে সেটা হয় একটা কম্পিউটারের খোলস, কম্পিউটার নয়। কম্পিউটার হল আপনার সামনের সেই আস্ত পিসি-টা যাতে আপনি লেখেন, গান বাজান, অ্যাকাউন্ট রাখেন, সিনেমা দেখেন, অঙ্ক কষেন, প্রোগ্রাম লেখেন, কম্পাইল করেন, যা দিয়ে আপনি ইন্টারনেটে ওয়েবসাইট থেকে ওয়েবসাইট চেখে বেড়ান মানে ব্রাউজ করেন, মেল লেখেন দূরে থাকা আপনজনকে — সেই জ্যান্ত জিনিষটা হল কম্পিউটার। সেই যন্তরটা হল ওই হার্ডওয়ার এবং সেই হার্ডওয়ারকে মিলিয়ে সেই হার্ডওয়ারের উপর সক্রিয় একটা সফটওয়ার ব্যবস্থা।

ক্রমে ক্রমে আমরা আরো ভালো করে বুঝব, প্রথমে ভাবা যাক, এই সফটওয়ারটা কী? একদম সরাসরি একটা মেশিন থেকে শুরু করি আসুন। আপনি মেশিনটা অন করলেন। স্ক্রিনের উপর দিয়ে অনেক অনেক লাইন অক্ষর ভেসে গেল। এবং শেষ অব্দি ফুটে উঠল হয় একটা ছবি — তার মধ্যেও কিছু লেখা আছে, নয়তো নিতান্তই কয়েক লাইন লেখা। যদি এটা একটা উইনডোজ বাতাবরণ হয় তাহলে ফুটে উঠল উইনডোজ ৯৮ কি ২০০০ কি এক্সপি, এবং তার সঙ্গে তার লোগোর ছবিটা। আর গ্নু-লিনাক্স হলে সেখানে সম্ভাব্য অনেকগুলো রাস্তা আছে, এক, সেটা কনসোল মোড হতে পারে, মানে কালো স্ক্রিনের উপর সারি সারি অক্ষর, তাতে লেখা থাকবে 'ম্যানড্রেক ৯' কি 'সুজে ৮.২' কি 'ডেবিয়ান ৩' কি 'রেডহ্যাট ৯' কি এইরকম কিছু, বা, গ্নু-লিনাক্স-এ নিজের ইচ্ছেমত দুইরকম ব্যবস্থাই করা যায় — আপনি চাইলে এই কনসোল মোডে না ফুটে উঠে উঠবে ওই উইনডোজ-এর মতই, ছবিসহ, লোগোসহ — এর নাম গুই মোড (GUI — **G**raphical-**U**ser-**I**nterface)। পরে অনেক বিশদ ভাবে আলোচনা আছে গুই-এর, এখন এটুকু মাথায় রাখুন যে, কনসোল মোডে কাজ করতে হয় লিখিত আদেশ কম্পিউটারকে কিবোর্ড থেকে টাইপ করে, তারপর এন্টার চাবিটা মেরে, মানে সেই নির্দেশটাকে কম্পিউটারে এন্টার করিয়ে। এন্টার হল মানে আদেশটা কম্পিউটারে পৌঁছল, সে কাজ শুরু করল।

আর, কনসোল মোডে এই টাইপ করে কাজ করার একটা বিকল্প ব্যবস্থা তৈরি হয়েছিল, কম্পিউটার মানবসমাজে কাম মানবজীবনে আসার বেশ কিছু বছর পরে, যেখানে আদতে কাজটা হয় দুটো স্তরে। একটা আসল স্তর যা গোপন রয়ে যায় আপনার কাছে, ব্যবহারকারীর কাছে। প্রকাশ্য স্তরে আপনি দেখেন একটা ছবি, ছবির গায়ে সুইচ, সেই সুইচের গায়ে মাউস দিয়ে ক্লিক করলেই কাজ শুরু হয়। মাউসের সঙ্গে কখনো কখনো অবশ্য ফাইলের নাম, ডিরেক্টরির নাম জাতীয় ছোটখাট জিনিষ কিবোর্ড দিয়ে টাইপ করেও ঢোকাতে হয়। মানে, আদেশের মূল মাধ্যমটা এখানে আর কিবোর্ড নয়, এখানে আপনি আদেশ পাঠাচ্ছেন মাউস দিয়ে। এই প্রকাশ্য স্তরের নিচে কাজ করছে একটা অপ্রকাশ্য স্তর যেখানে আপনার এই মাউস-ইভেন্ট মানে ক্লিক করার ঘটনাটাকে কম্পিউটারের অপারেটিং সিস্টেম অনুবাদ করে নিচ্ছে কনসোল মোডের আদেশের মত অক্ষর এবং শব্দভিত্তিক কমান্ডে, এবং সেই কমান্ড অনুযায়ী কাজ করে চলেছে। যেরকম আদেশ আপনি নিজেই দিতেন যদি আপনি কনসোল মোডে বা কমান্ড মোডে কাজ করতেন। এখন আপনি তা করছেন, ছবি এবং মাউস দিয়ে এই কাজ করাটাই হল গুই মোড।

কনসোল হোক আর গুই হোক যেকোনো একটা মোডে আপনি কম্পিউটারে ঢুকলেন। এবার আপনি যে কাজটা চান সেটাকে আপনার চালু করতে হবে। ধরুন আমরা তিনটে কাজ ভাবি, লেখা, গান শোনা, এবং প্রোগ্রাম লেখা আর চালানো। প্রথম কাজটা ওয়ার্ড প্রসেসরের। গ্নু-লিনাক্স হলে এই প্রোগ্রামটার নাম ওপন-অফিস বা কে-অফিস বা অ্যাবিওয়ার্ড বা অন্য কিছু। লিনাক্সে সমস্যাটা দারিদ্রের নয়, প্রাচুর্যের। যে কোনো কাজের জন্যেই এত এত প্রোগ্রাম তৈরি হয়ে আছে যে প্রথম দিকে অনেক ব্যবহারকারীর কাছে সেটা একটা গোলকধাঁধার মত লাগে। প্রতিবেশটা উইনডোজ হলে, ওয়ার্ডপ্রসেসিং বা শব্দ-চটকানোর জন্যে আপনি ব্যবহার করছেন এমএস ওয়ার্ড বা অ্যাডোব পেজমেকার বা লোটাস স্মার্টশ্যুট রাইটার।

গান শোনার জন্যে আপনি গ্নু-লিনাক্সে ব্যবহার করতে পারেন এমপিজি১২৩ বা এমপিজি৩২১ বা ফ্রি-অ্যাম্প বা এক্সএমএমএস বা এমপ্লেয়ার বা এরকম অজস্র কিছুর একটা। উইনডোজ হলে যেখানে ব্যবহার করতেন উইনডোজ মিডিয়া প্লেয়ার বা উইনঅ্যাম্প।

আর প্রোগ্রাম লেখাটাও লেখা, কিন্তু তার জন্যে ওয়ার্ড প্রসেসরে কাজ হয়না, ওয়ার্ড প্রসেসর কাজ করে আপনার লেখার ফর্মাটিং নিয়ে। আর প্রোগ্রামগুলো লেখা হয় সাদামাটা বা প্লেন টেক্সটে, সেখানে কোনো ফর্মাটিং থাকেনা। ফন্ট বা অক্ষর কী হবে, প্যারাগ্রাফের আগে পরে ডানে বাঁয়ে জায়গা ছাড়া হবে কিনা, ইত্যাদি সবকিছুই এক কথায় হল ফর্মাটিং। শূন্য নম্বর দিনের ফর্মাটিং নিয়ে আলোচনা মনে করুন। ফর্মাটিংরিক্ত নিখাদ নির্জলা টেক্সট লেখার এডিটরগুলোকে সচরাচর ডাকা হয় কমান্ড এডিটর বলে। গ্নু-লিনাক্সে কমান্ড এডিটর আছে ইম্যাক্স ভিম জো জেডিট নেডিট গেডিট কে-রাইটার জাতীয় গুচ্ছ গুচ্ছ সফটওয়ার। প্রোগ্রামিং করা মানে কিন্তু শুধু প্রোগ্রাম লেখা না। প্রোগ্রাম লেখা শেষ হওয়ার পরে তাকে কম্পাইল করতে হয় কম্পাইলার দিয়ে। এগুলোর সঠিক অর্থ কী তাই নিয়ে এখুনি মাথা ঘামাবেন না, সময় হলেই আপনি জেনে যাবেন, জানতে হবে আপনাকে। এই কম্পাইল করার কাজের জন্যে গ্নু-লিনাক্সে আছে গ্নু-কম্পাইলার-কালেকশন বা জিসিসি। বলতে মাত্র একটা বাক্য লাগল, কিন্তু এটাই আসলে গোটা গ্নু-লিনাক্স আন্দোলনের নিষেকভূমি, বিরাট একটা ইতিহাস আছে এখানে, যার একটু আমরা জানব পাঁচ নম্বর দিনের শুরুতে। আর উইনডোজ হলে লেখার জন্যে কমান্ড এডিটর হিশেবে নোটপ্যাড আর কম্পাইল করার জন্যে বোরল্যান্ড মাইক্রোসফট বা ইনটেল ইত্যাদি কম্পাইলার, নানা প্রোগ্রামিং ভাষায় লেখা প্রোগ্রাম কম্পাইল করার।

প্রোগ্রাম লেখা আর কম্পাইল করার আর সেই কম্পাইল হওয়া প্রোগ্রাম চালানোর কাজগুলোকে আলাদা আলাদা ভাবে না করে, একটা অ্যাপ্লিকেশনের মধ্যেও করার ব্যবস্থা আছে। তাকে বলে আইডিই (IDE — **I**ntegrated-**D**evelopment-**E**nvironment) গোছের জিনিষগুলোকে আনলাম না। উইনডোজ বা গ্নু-লিনাক্স দু ধরনের ব্যবস্থাতেই আছে। সেগুলোও আপনি ব্যবহার করতে পারেন প্রোগ্রামিং এর কাজে। যাই হোক, এমন অবস্থা ভাবুন, যেখানে, গ্নু-লিনাক্স হোক বা উইনডোজ আপনি এই তিনটে কাজ করছেন, ওয়ার্ড-প্রসেসিং, গান-শোনা, এবং প্রোগ্রামিং। এই তিনটে কাজ আপনি চালাচ্ছেন আপনার মেশিনে, আপনার সিস্টেমে। এর পুরো প্রক্রিয়াটাকে একটু বোঝার চেষ্টা করা যাক। আপনি এই সময়ে কী কী করছেন, এবং আপনার মেশিনই বা কী কী করছে।

ওয়ার্ড প্রসেসর মানে ফর্মাটিং সহ লেখার জন্যে আপনি যে প্রোগ্রামই ব্যবহার করুন, যে বাতাবরণই হোক, উইনডোজ বা গ্নু-লিনাক্স — এবার আপনাকে টাইপ করে নিজের লেখাটাকে লিখে যেতে হবে এবং প্রয়োজন মোতাবেক মাউস বা কিবোর্ড ব্যবহার করে সেখানে নিজের ইচ্ছামত কাজ করতে হবে, বদল আনতে হবে। ধরুন, পরপর তিনটে লাইনের মোট লেখাটা আপনি গাবদা গাবদা গরুপাঠ্য সাইজের বর্ণমালায় দেখতে চান, তার জন্যে আপনাকে মাউস দিয়ে বা কিবোর্ড দিয়ে গোটাটাকে সিলেক্ট করতে হবে, এবং মাউস বা কিবোর্ড দিয়ে আদেশ দিয়ে তাদের ফন্ট সাইজ বাড়াতে হবে। আবার একটা শব্দ যোগ করতে হলে সেটা করতে হবে কিবোর্ড দিয়ে। খুব বিশেষ রকমে এমনকি সেই কাজটাও যদিও মাউস দিয়েই করা যায়। সেই রকম অ্যাপ্লিকেশন আছে যা স্ক্রিনে একটা কিবোর্ডের ছবি ফুটিয়ে তোলে, এবং সেখানে যে অক্ষরের ছবিতে আপনি ক্লিক করবেন, আপনার লেখায় সেটা যোগ হয়ে যাবে। বা গান শুনতে হলে আপনাকে গানের প্রোগ্রামটা খুলে কোন গান আপনি বাজাতে চান সেই ফাইলটাকে তার ডাইরেক্টরিতে আপনাকে খুঁজে দিতে হবে। সেটাও করবেন মাউস বা কিবোর্ড দিয়ে। গানটা বাজতে শুরু করবে। বা কম্পাইল করছেন যখন প্রথমে আপনাকে প্রোগ্রামটা লিখে সেভ করতে হবে একটা ফাইলনাম দিয়ে, তারপর

তাকে কম্পাইল করতে বলতে হবে। তার মানেও সেই মাউস আর কিবোর্ড। কম্পাইল করা মানে কী সেটা আমরা একটু বাদেই জানব, উতলা হবেন না।

এই প্রতিটি ক্ষেত্রেই আপনি কী করছেন — হয় আপনার ইচ্ছের প্রোগ্রামের নাম লেখা ছবি দেওয়া আইকনে ক্লিক করছেন, মানে মাউস দিয়ে আদেশ দিচ্ছেন, প্রোগ্রামটা চালু হচ্ছে, অথবা কিবোর্ড থেকে আদেশ দিচ্ছেন টাইপ করে। কী আদেশ দিচ্ছেন? অমুক নামের প্রোগ্রামটা চালু করো। এই প্রোগ্রামটা হল একটা প্রয়োগ — অ্যাপ্লিকেশন। এরা কম্পিউটারের গোটা ব্যবস্থার একদম উপরের স্তর। যে কোনো একজন ব্যবহারকারী সরাসরি মুখোমুখি হয় এই অ্যাপ্লিকেশনগুলোর। এই অ্যাপ্লিকেশনগুলো কিছু ক্ষেত্রে একটা একক প্রোগ্রাম হতে পারে, অনেকক্ষেত্রেই একাধিক প্রোগ্রামের একটা সমাহার। যেমন ওপন-অফিস হোক বা এমএসঅফিস, লিখতে লিখতে আপনি একটা ছবি ঢোকাতে চাইলেন বা একটা সমীকরণ, ফন্ট মানে অক্ষরের চেহারা বদলাতে চাইলেন, বা সরাসরি এইখান থেকেই একটা মেল করতে চাইলেন কাউকে — এই প্রত্যেকটা কাজের সময়েই পর্দার পিছনে মূল প্রোগ্রামটা, মানে ওপন-অফিস বা এমএসঅফিস, ডেকে নিল নানা ধরনের নানা প্রোগ্রামকে। এই সমস্ত প্রোগ্রামগুলোকে মিলিয়েই হল গোটা প্রয়োগ বা অ্যাপ্লিকেশনটা।

এবার ভাবুন এই প্রোগ্রামগুলো কী? এগুলো আসলে ফাইল। সেখানে পরপর লেখা আছে আদেশ। কিন্তু আদেশগুলো এমন ভাষায় লেখা যা আপনি আমি বুঝব না, কিন্তু কম্পিউটার বুঝবে। যার নাম মেশিন ল্যাংগোয়েজ। দিন শূন্যর সেই ভিশুয়াল চ্যাংড়ামোটা মনে করুন, যেখানে আমরা একটা পিডিএফ মানে বিশেষ ফর্মাটিং-এর ফাইলকে কোনো একটা কমান্ড এডিটর দিয়ে খুলেছিলাম। নিছক ফাইলটাই কিন্তু প্রোগ্রাম নয়, বা যে কোনো ফাইলও প্রোগ্রাম নয়। সেই সব ফাইলেরাই প্রোগ্রাম যারা চলতে পারে। সেই ফাইল যা নিজে চলতে পারে, সে যখন চলে সেটা হল ওই নামের প্রোগ্রামের একটা প্রসেস বা প্রক্রিয়া। এগুলো সবই খুঁটিয়ে খুঁটিয়ে বুঝতে হবে আমাদের। নানা রকম জটিলতা নিয়ে ধীরে ধীরে। এখন শুধু একটা প্রাথমিক ধারণা করে নিচ্ছি আমরা। যা বলছিলাম, সব ফাইল নিজে চলতে পারেনা। ধরুন, আপনার ওয়ার্ডপ্রসেসরে আপনি একটা প্রবন্ধ লিখলেন, প্রবন্ধটা নিজের পছন্দমতন একটা নাম দিয়ে সেভ করলেন মানে তুলে রাখলেন কম্পিউটারের কোনো ডিস্কে, কোনো স্মৃতির ভাঁড়ারে। এবার হার্ডডিস্কিত ওই প্রবন্ধটা একটা ফাইল, কিন্তু নিজে নিজে চলেনা। আবার আপনার ওয়ার্ড-প্রসেসরটাও একটা ফাইল যা নিজে নিজে চলে। নিজে নিজেও চলে, আবার ওই প্রবন্ধের নাম করে আদেশ দিলে প্রবন্ধটাও খুলে দিতে পারে, তাকে আপনি আর একবার ওয়ার্ডপ্রসেস করতে পারেন, শব্দ-চটকাতে পারেন। এই আদেশটা আপনি মাউস দিয়েও দিয়ে থাকতে পারেন, ফাইল মেনুতে গিয়ে ওপন সাবমেনু ক্লিক করলে সে হার্ডডিস্কের ভিতর ফাইল আর ডিরেক্টরির কাঠামোটা খুলে দেয়, সেখানে প্রবন্ধটা খুঁজে দিতে হয়। আবার সরাসরি কিবোর্ড দিয়েও দেওয়া যেত আদেশটা।

এবার নিজে নিজে চলা এই ফাইলগুলো মানে প্রোগ্রামগুলোকে মিলিয়ে মিলিয়েই তৈরি হয় অ্যাপ্লিকেশন। এই চলমান প্রোগ্রাম-সমাহার অ্যাপ্লিকেশনগুলোর বন্যের বন এবং শিশুর মাতৃক্রোড় হল আপনার কম্পিউটারের সামগ্রিক প্রতিবেশটা, মানে আপনার মেশিনের অপারেটিং সিস্টেম। ধরুন ম্যানড্রেক ৯.১ কি সুজে ৮.২ কি উইনডোজ ২০০০ — এরা হল অপারেটিং সিস্টেম। এরকম আরো আছে ইউনিক্স, সোলারিস, আইক্স ইত্যাদি। এই অপারেটিং সিস্টেমও এক রকমের প্রোগ্রাম। তার মানে একটা ফাইল বা একাধিক ফাইলের একটা সমাহার। যে ফাইল নিজে চলে এবং অন্য সব প্রোগ্রামকে চালায়। অপারেটিং সিস্টেম নামক প্রোগ্রামটা আবার শুধু অন্য ফাইলদেরই চালায় না, চালায় হার্ডওয়ারগুলোকেও, অন্য প্রোগ্রামদের হার্ডওয়ার ব্যবহারের ব্যবস্থা করে দেয়। একটা চলমান কম্পিউটারে সবচেয়ে প্রাথমিক, সবচেয়ে শক্তিশালী এবং সবচেয়ে জরুরি প্রোগ্রাম এই অপারেটিং সিস্টেম। আমরা ক্রমে আরো বিশদ জানব এদের নিয়ে।

২.১।। ভৌত উপাদান — কম্পিউটার সিস্টেমের ভিত

তাহলে একটা জ্যান্ত কম্পিউটারের উপাদান কী কী? একদিকে গুচ্ছ গুচ্ছ হার্ডওয়ার। সিপিইউ, র‍্যাম, রম, হার্ডডিস্ক, ফ্লপি ইত্যাদি। এর সঙ্গে আছে গোটা প্রোগ্রাম জগতটা — অপারেটিং সিস্টেম থেকে অ্যাপ্লিকেশন। এরা আবার কাজ করে ওই হার্ডওয়ার উপাদানগুলোর উপরেই, এদের ব্যবহার করে, এদের উপর, এদের দিয়ে। এদের এক এক পিস এক এক জাতের জিনিষ, কারুর লেজ নেই, কারুর বা উনিশটা লেজ, কেউ আবার লেজ কাকে বলে তাই বোঝে না।

তাহলে এই বহুমুখী বিচিত্রমুখী এইসব বিদঘুটেপনাকে একত্রিত করে মিলিয়ে একটা চলমান কম্পিউটারের নিখুঁত সমগ্রে নিয়ে আসছে কে?

এই সমস্ত উপাদানকে যথাযথভাবে নিয়ন্ত্রণ এবং ব্যবহার করার কাজটা কিন্তু আদৌ সহজ না। ইন ফ্যাক্ট, সাইবার জগতে এটাই সবচেয়ে খিঁটকেল কাজ। এটাই সেই ভিত, যার উপর দাঁড়িয়ে রয়েছে আর সব কিছু, সমস্ত কিছু। এই কাজটা সহজ নয়, তাই এই কাজ করার ক্যালিসম্পন্ন প্রোগ্রাম লেখার কাজটাও একটা সহজ নয়। জটিলতার একটা খুব দুধেভাতে জাতের আন্দাজ নেওয়ার চেষ্টা করা যাক। ধরুন আপনি আপনার ওয়ার্ড প্রসেসরে যে প্রবন্ধটা লিখছেন, সেটায় কিছু বদল আনলেন, এনে একটা অন্য নামে সেভ করলেন। তারপর তার একটা প্রিন্ট-আউট নিলেন। এখানে অপারেটিং সিস্টেমের কাজের একটা আলগা এবং উপর-উপর তালিকা বানানো যাক।

এর মানে আপনার আদেশ অনুযায়ী ওয়ার্ড প্রসেসর নামক প্রোগ্রামটা ব্যবহার করল সিপিইউকে, যে মূল কাজটা করেছে। সিপিইউ যা করে, কম্পিউটারে কেন্দ্রীয় ভূমিকাটাই তার, সব কাজই তার। তাকে কাজে লাগাল প্রোগ্রামটা। তার মানে, এই প্রোগ্রামের হয়ে সিপিইউকে কাজে লাগাল অপারেটিং সিস্টেম বা কারনেল। আগেই তো বলেছি, এই হার্ডওয়ার উপাদানগুলোর সরাসরি নিয়ন্ত্রণ একমাত্র সব প্রোগ্রামের নিয়ন্তা প্রোগ্রাম মানে কারনেলের, সাধারণ প্রোগ্রামের কোনো রাইটই নেই হার্ডওয়ারের সঙ্গে সরাসরি কথা বলার।

শুধু সিপিইউ নয় কিন্তু, কাজে লাগানো হয়েছে মেমরিকেও। অন্য নামে ফাইলটাকে সেভ করার আগে মেমরির আলমারির তাকে তুলে নিতে হয়েছে গোটা ফাইলের মোট তথ্যটাকে। ওই প্রবন্ধের ফাইলটার অন্তর্বস্তু নিয়ে কিছু করা মানেই তো ফাইলের আভ্যন্তরীন তথ্যটাকে তুলতে তুলতে নামাতে নামাতে যাওয়া। মেমরিকে কাজ করতে হচ্ছে নিরন্তরই। হার্ডডিস্কে হোক কি যেখানে হোক এক বাইট তথ্য নামানো বা ওঠানো মানেই, আগেই সেই বাইটটাকে মেমরিতে তুলে নেওয়া।

একাধিক বার একাধিক ভাবে কাজে লাগাতে হয়েছে হার্ডডিস্ককে। প্রথমে, হার্ডডিস্কের ভিতরের সেই থালা বা প্ল্যাটারগুলোকে ঘুরিয়ে ঘুরিয়ে সেই সেই সেক্টরগুলো আনতে হয়েছে যেখানে মূল ফাইলটা লেখা ছিল, তাদের শরীরে লিখে রাখা তথ্যটাকে পড়ে তুলে নিতে হয়েছে মেমরির তাকে, তারপর নতুন নামে সেভ করার সময় আবার একবার চরকিপাক ঘোরাতে হয়েছে বেচারা হার্ডডিস্ককে, এবার তার শরীরে আবার কিছু নির্দিষ্ট সেক্টরে লেখা হয়েছে নতুন কিছু তথ্য — মানে নতুন নামের তথ্য। এর সঙ্গে সঙ্গে অগোচরে চলেছে আরো কিছু কাজ, হার্ডডিস্কের পার্টিশনে কী কী ডিরেক্টরিতে কী কী ফাইল বদলালো তার বদলে যাওয়া খুঁটিনাটি আবার লিখে রাখতে হয়েছে হার্ডডিস্কের মধ্যেই লিখে রাখা ফাইল-সংক্রান্ত তথ্যে, যার নাম মেটাডেটা। ডেটার বিষয়ে ডেটা। ভালো করে এটা বুঝব আট নম্বর দিনে গিয়ে।

কাজে লাগাতে হয়েছে আপনার কনসোলকে, যখন আপনি ফাইলকে বদলেছেন, ফাইলের প্রত্যেকটা বদলকে সে স্ক্রিনে ফুটিয়ে তুলেছে। আপনার প্রোগ্রাম অ্যাপিল করেছে অপারেটিং সিস্টেম বা কারনেলের কাছে, কারনেল কথা বলেছে স্ক্রিন নামক ভৌত উপাদানের কন্ট্রোলারের সঙ্গে, কন্ট্রোলার সরাসরি তথ্য পাঠিয়েছে স্ক্রিন বা কনসোলকে।

কাজে লাগানো হয়েছে প্রিন্টারকে, যখন সে প্রিন্ট নিয়েছে। আবার ওই প্রোগ্রাম থেকে কারনেল থেকে প্রিন্টার-ড্রাইভার থেকে প্রিন্টার-কন্ট্রোলার থেকে প্রিন্টার। ফটফটে সাদা কাগজে খটখটে কালো বর্ণমালায় ফুটে উঠেছে আপনার প্রবন্ধ। চাইলে ইস্টম্যানকালার প্রবন্ধও লিখতে পারেন, সরকারি কোনো নিষেধ নেই।

উপরের এই তালিকাটা কিন্তু অতিসরলীকৃত বললেও কম বলা হয়। কারণ, প্রতিটি মুহূর্তে আপনার চালানো প্রোগ্রামের পাঠানো আর্জি রাখতে গিয়ে কারনেলকে মাথায় রাখতে হয়েছে আপনার গোটা মেশিনের প্রতিটি হার্ডওয়ার ডিটেইলস, আপনার মাদারবোর্ডের সার্কিটে বিদ্যুৎপ্রবাহের পারমানবিক ওঠানামা, প্রতিটি আইসির পিনের সংখ্যা, মেশিনের হার্ডডিস্কের প্রতিটি প্ল্যাটারের প্রতিটি ডিটেইলস — উঃ, চাইলে এই তালিকাটাকে আরো বহু সময় ধরে বাড়িয়ে চলা যায়, প্রায় অ্যাড ইনফিনিটাম।

এই গোটা ঘটনাটা ঘটল কী করে? কে এই গোটা বহুমুখী প্রক্রিয়াটাকে নিয়ন্ত্রণ করল। আপনার ওয়ার্ড প্রসেসর প্রোগ্রামের ভূমিকাটুকু খেয়াল করুন। তার আর্জি মোতাবেক ঘটেছে এই প্রতিটি ঘটনা। ওয়ার্ড প্রসেসর প্রোগ্রামটার ভিতরেই সেই ব্যবস্থা করা ছিল। তার শরীরে ইঁদুর ছেড়ে দিয়ে বা কিবোর্ড ঠুকে আপনি যখনি কোনো আব্দার

জানাবেন, তার সঙ্গে তাল মিলিয়ে ঠিক কী আর্জি জানাতে হবে কারনেলের কাছে। যখনই আপনি প্রিন্ট করতে বলবেন, আপনার ওয়ার্ড প্রসেসর জানে, কোন জায়গায় কোন ভাবে কী আবেদন জানাতে হবে, বাবা কারনেলের চরণে সেবা লাগি, এই এই তথ্য এই এই দিকে, প্রিন্টারের পোর্টের দিকে পাঠিয়ে দাও বাবা, যাতে সেটা কেবল বেয়ে কেবল প্রিন্টারের কাছেই পৌঁছয়, এবং আপনি ছাপার আখরে নিজের লেখা পড়ে মুগ্ধ হতে পারেন।

অর্থাৎ হার্ডওয়ারের উপাদানগুলোকে তার কাজে লাগাতেই হবে। শুধু তার কেন, যে কোনো প্রোগ্রামেরই। প্রোগ্রামগুলোর কাজ কী? চলা, চলে কোনো একটা কাজ করা। কাজটা কোথায় হয়? হার্ডওয়ারে। এবার এই বিভিন্ন হার্ডওয়ার উপাদান, তাদের এক এক জনের এক এক রকম খুঁটিনাটি। কেউ তৈরি চৌম্বক পদার্থে, কেউ হল একটা পিকচার টিউব, কেউ আবার একটা নড়তে থাকা কাঁটা, পিছনে একটা কালির দোয়াত। তাদের প্রত্যেকটা খুঁটিনাটিকে মিলিয়েই তাকে কাজ করতে হবে। সেই কাজের সুবিধের জন্যে এই সমস্ত উপাদানের উপর সর্বব্যাপী একটা প্রলেপের মত ছড়ানো থাকে সফ্টওয়ারের একটা স্তর — এটারই সিরিয়াস নাম অপারেটিং সিস্টেম। প্রোগ্রামগুলোকে একটা সহজ ব্যবহারযোগ্যতা দেওয়াই যার কাজ। 'ফ্লপিতে লেখো', 'মোডেমে সংযোগ করো' বা 'হার্ড-ডিস্কের এই ফাইল উড়িয়ে দাও' — আপনি এই আদেশ দিলেই আপনার প্রোগ্রামের হয়ে এই অপারেটিং সিস্টেম চট করে বুঝে নিতে পারে 'ফ্লপি' বা 'মোডেম' বা 'হার্ড-ডিস্ক' কী ধরনের বস্তু কিম্বা 'লেখা' বা 'সংযোগ' বা 'ওড়ানো' মানেই বা কী।

এই তালিকায় আমরা কম্পিউটার হার্ডওয়ার থেকে সফটওয়ারের স্তরবিন্যাসটা দেখিয়েছি। শুধু একটা জিনিষ খেয়াল করার — স্তরগুলো কিন্তু নিচ থেকে উপরে। ভিত-এর উপর বাড়ির মত।

<table>
<tr><th colspan="4">হার্ডওয়ার থেকে সফটওয়ার স্তরবিন্যাস</th></tr>
<tr><td>(৬) হিসাবনিকাশের সফটওয়ার</td><td>(৬) রেলের টিকিট বিলিব্যবস্থা</td><td>(৬) ইন্টারনেটে ঘোরাঘুরি</td><td>প্রায়োগিক সফটওয়ার বা অ্যাপ্লিকেশনের এলাকা</td></tr>
<tr><td>(৫) কম্পাইলার</td><td>(৫) এডিটর</td><td>(৫) কমান্ড ইন্টারপ্রিটার</td><td rowspan="2">মূল কাঠামো বা সিস্টেম প্রোগ্রামের এলাকা</td></tr>
<tr><td colspan="3">(৪) অপারেটিং সিস্টেম</td></tr>
<tr><td colspan="3">(৩) মেশিন ল্যাংগুয়েজ</td><td rowspan="3">হার্ডওয়ার বা ভৌত যন্ত্রপাতির এলাকা</td></tr>
<tr><td colspan="3">(২) মাইক্রোআর্কিটেকচার</td></tr>
<tr><td colspan="3">(১) ভৌত উপাদানগুলো</td></tr>
</table>

এক নম্বর হল হার্ডওয়ার-এর এলাকা, এর মধ্যে তিনটে স্তর। ভৌত উপাদান, মাইক্রোআর্কিটেকচার, মেশিন ল্যাংগুয়েজ। এর পরের এলাকা মূল কাঠামো বা সিস্টেম প্রোগ্রামের — মেশিনের অপারেটিং সিস্টেম কী ভাবে চলে। এর মধ্যে স্তর আবার দুটো। এক নম্বরে একা অপারেটিং সিস্টেম মানে ৪ নম্বর। উপরেরটায় মানে পাঁচ নম্বর স্তরে তিনটে অংশ — কমান্ড ইন্টারপ্রিটার, এডিটর এবং কম্পাইলার। একদম উপরের স্তরটা মানে ৬ নম্বর হল প্রায়োগিক সফটওয়ার বা অ্যাপ্লিকেশন। এর মধ্যে আবার নানা বিচিত্র অংশ আছে। আমরা এখানে তিনটে উদাহরণ দিয়েছি। ছকটা মাথায় তুলে ফেলুন, ধীরে ধীরে প্রয়োজনীয় খুঁটিনাটিগুলো স্পষ্ট হয়ে যাবে পাঠমালাটা চলতে চলতে। আপনাকে মাঝে মাঝেই ফেরত আসতে হবে পুরোনো দিনগুলোয়। এখানে স্তর ভাগ ছটা — একদম প্রাথমিক বা নিচের স্তরটা হল ভৌত উপাদানের মানে সার্কিট প্লাস্টিক ধাতু সিলিকনের। আপনার পিসির টাওয়ারের গায়ের ক্যাবিনেটটা খুললে মধ্যে আপনি যে জগতটা পাবেন। মাদারবোর্ড, এক্সপ্যানশন স্লট, এক্সপ্যানশন কার্ড, সার্কিট বোর্ড, অ্যাডাপ্টার, ট্রানজিস্টর, ক্যাপাসিটর, তার, তামা, রেজিস্টান্স, কানেক্টর, হার্ডডিস্ক-এর কৌটো, সিডি-ড্রাইভের কৌটো, ফ্লপি-ড্রাইভের কৌটো, দু-চারটে ফ্যান — এটা মূলত ইলেকট্রিকাল আর ইলেকট্রনিক এঞ্জিনিয়রের এলাকা। এই ১ নম্বর ভৌত উপাদানের স্তরের ঠিক উপরে উপরে দাঁড়িয়ে আছে মাইক্রোআর্কিটেকচার আর মেশিন ল্যাংগুয়েজের ২ আর ৩ নম্বর স্তর। এই ১, ২ আর ৩ নম্বর স্তরকে মিলিয়ে হার্ডওয়ার বা ভৌত উপাদানের এলাকা। দেখুন আমরা ডানদিকে দেখিয়েছি। এর উপরের এলাকাটা হল মূল কাঠামোটাকে ঘিরে প্রোগ্রামের বা সিস্টেম প্রোগ্রামের এলাকা, দেখুন ডানদিকে দেখানো।

তার মধ্যে আছে দুটো স্তর। ৪ নম্বর গোটা স্তরটা জুড়ে একা কুম্ভ অপারেটিং সিস্টেম, মাদার মাস্তান অফ অল মাস্তানস। তার উপরে ৫ নম্বর স্তরটায় আমরা তিনটে উপাদান দেখিয়েছি — কম্পাইলার এডিটর এবং কমান্ড ইন্টারপ্রিটার। এডিটর দিয়ে প্রোগ্রাম লেখা হয়, বা কনফিগারেশন টেক্সট ফাইল। এই এডিটর হল বাহুল্যবর্জিত ফর্মাটিং বর্জিত টেক্সট এডিটর, দেখানোর আর খাওয়ার দুটো আলাদা নয়, একটাই মাত্র দাঁত। কম্পাইলার দিয়ে লেখা প্রোগ্রামকে কম্পাইল করে চালানোর মত ফাইল বা এক্সিকিউটেবল প্রোগ্রাম ফাইল বানানো হয়। আর কমান্ড ইন্টারপ্রিটার হল আমাদের আর কম্পিউটারের মধ্যে দোভাষি। কোনো চালু প্রোগ্রাম মারফত বা সরাসরি সিস্টেমকে কোনো আদেশ যখন দিই আমরা, সেটার জন্যে সিস্টেমকে ঠিক কী করতে হবে — এটাই সিস্টেমকে বুঝিয়ে দেয় কমান্ড ইন্টারপ্রিটার। গ্নু-লিনাক্সে সবচেয়ে জনপ্রিয় কমান্ড ইন্টারপ্রিটার হল ব্যাশ। পরে এই নিয়ে গুষ্টি গুষ্টি পড়তে হবে আমাদের। আর এই ৫ নম্বর স্তরের উপরের গোটাটাই হল ব্যবহারকারীর ব্যবহৃত প্রায়োগিক বা অ্যাপ্লিকেশন সফটওয়ারের এলাকা — যে ৬ নম্বর স্তরে আমরা তিনটে প্রয়োগ সফটওয়ারের উদাহরণ দেখিয়েছি, হিশেবনিকেশ, রেলের টিকিট আর ইন্টারনেট ব্রাউজিং। উদাহরণ কিন্তু অজস্র হতে পারে। আমাদের চারপাশে সফটওয়ার বলতে আমরা সচরাচর যা বুঝি তার প্রায় গোটাটাই এই ৬ নম্বর স্তরের মধ্যে সীমাবদ্ধ।

এই স্তরের মধ্যে তিনটে তার উপরেরটা হল মূল কাঠামো বা সিস্টেম প্রোগ্রামের এলাকা। তার মধ্যে দুটো উপস্তর। একটা উপস্তর অপারেটিং সিস্টেমের একার। অন্যটা আর একদম উপরেরটা হল প্রায়োগিক সফটওয়ারের, মানে বেশিরভাগ ব্যবহারকারীর, এখানে তিনটে উদাহরণ দেওয়া হয়েছে। আরো কোটি কোটি আছে। সবচেয়ে বেশি কম্পিউটার এখন যেভাবে ব্যবহার হয় — বাজনা-টিভি-ব্রাউজারের থ্রি-ইন-ওয়ান — সেটাও এই ছয় নম্বর স্তরে। এবার আজকের আলোচনার অবশিষ্টটা জুড়ে আমরা কম্পিউটারের সফটওয়ারের এই এলাকাগুলো একটু ভালো করে বুঝব।

২.২।। মাইক্রোআর্কিটেকচার আর মেশিন ল্যাংগুয়েজ — হার্ডওয়ার থেকে সফটওয়ার

সিলিকন প্লাস্টিক ধাতু ইত্যাদি দিয়ে তৈরি ভৌত উপাদানগুলোর স্তরের ঠিক উপরের দু-নম্বর স্তরটাই হল মাইক্রোআর্কিটেকচারের। এখানে একদম কাঁচা আছোলা ভৌত উপাদানগুলোকে মিলিয়ে মিলিয়ে এক একটা কেজো ইউনিট বানানো হয়। এর কেন্দ্রীয় জায়গায় দাঁড়িয়ে অর্কেস্ট্রার কন্ডাক্টরের মত ব্যাটন দোলায় সিপিইউ চিপ। সিপিইউ মানে মূল চিপ এবং তার শরীরে দেগে দেওয়া ওই রেজিস্টারগুলো, একটু আগে যাদের কথা আমরা বললাম।

সিপিইউ-র ভিতর থাকে নানা ভৌত উপাদান — থাকে আলু মানে অ্যারিথমেটিক লজিক ইউনিট এবং অন্যান্য অংশ। এবং থাকেওই আলুর ভিতরে বাইরে অবশিষ্ট সিপিইউর সঙ্গে তার তথ্য বা ডেটা চলাচলের সংযোগ বা বাস। প্রতি নির্দিষ্ট সময় একককে নির্ভুল ভাবে নিরবচ্ছিন্ন ভাবে সূচিত করে চলার জন্যে, যাকে কম্পিউটার বিজ্ঞানে ডাকা হয় ক্লক-সাইকেল বলে, প্রয়োজনীয় সার্কিটও থাকে ওই সিপিইউ-র ভিতরেই, আমরা আগেই বলেছি।

সিপিইউ-র ওই রেজিস্টারগুলো থেকে সূচক তোলা হয় আর অ্যারিথমেটিক-লজিক ইউনিটে তাদের পাক দেওয়া হয় — যেমন, দুটো সংখ্যাকে যোগ করা বা বুলিয়ান অ্যান্ড (AND)। ফলাফলটা আবার তুলে দেওয়া হয় ওই রেজিস্টারের শিকেয়। ধরা যাক একটা হার্ড ডিস্কের একটা নির্দিষ্ট সেক্টর থেকে একটা সংখ্যাকে পড়া হল, আর একটা সেক্টর থেকে আর একটা সংখ্যা, তাদের যোগ করা হল, মানে ওই অ্যান্ড, তারপর সেই যোগফলটাকে ফের দেগে দেওয়া হল হার্ড ডিস্কের গায়ের আর একটা সেক্টরে। বাস্তবে জাটিল্যের সংখ্যা আর একটু বেশি। আরো অজস্র নিয়ন্তা বা প্যারামিটার তাদের নিজের নিজের খেলার প্রতিভা দেখিয়ে থাকে এখানে, যার কিছু ইঙ্গিত আমরা ইতিমধ্যেই দিয়েছি। কিন্তু, মাইন্ড ইট, আমি আগেই বলেছি, এও অতি অল্প হইল। এবং ওই ড্রাইভের মানে হার্ড ডিস্কের ব্যাপারটাও রীতিমত খ্যাঁচালো। আর তথ্য বা ডেটা ঢোকানো-বেরোনোর মানে ইনপুট-আউটপুটের কায়দাও শুধু ওই সবেধন হার্ডমণি নয়, তাও তো জানেন। ধরুন আলুতে এই যোগ বিয়োগ কষাকষির ফলাফলটা শেষ অব্দি যাবে মোডেম মারফত সার্ভার। সার্ভার বলতে বোঝায়, শিশুপাঠ্য রকমে বললে, জংশন স্টেশনের মত বহুমুখী যোগাযোগসম্পন্ন কিঞ্চিৎ গাবদা গতরের একটা কম্পিউটার। এবার ভাবুন, কষাকষির আর লেখালেখির মধ্যে সময়সীমাটা যদি খুব বেড়ে যায় তাহলে মধ্যের অংশগুলো বেয়ে মোডেম বেয়ে সার্ভার অব্দি পৌঁছনোর আগেই, সার্ভারটাই হয়ত যোগাযোগের পথটা বন্ধ করে দেবে। তার মানে গোটা কাজটাই মিনিংলেস হয়ে যাচ্ছে। এই রকম অজস্র জটিলতা থাকে সেখানে। এর উপরের, সিস্টেম প্রোগ্রামিং-এর এলাকার থেকে অপারেটিং সিস্টেম যখন

কোনো কাজ করিয়ে নিচ্ছে সিপিইউকে দিয়ে, সেই সিপিইউ কিন্তু কাজ করছে তার চারপাশের এই ভৌত জটিলতাকে নিয়েই। তাই অপারেটিং সিস্টেম তথা গোটা সিস্টেম প্রোগ্রামিং-কেই খেয়াল রাখতে হয়, মাথায় রাখতে হয় এই জটিলতাগুলোকে। কোন কাজ করাতে হলে ভৌত উপাদানগুলোকে কী করতে হবে। সেখানে কতগুলো জিনিষ সম্ভব, কতগুলো জিনিষ অসম্ভব, ইত্যাদি। ভৌত উপাদানের জগতের কাছে কোনো অন্যায় আব্দার জানালে চলবে না, সে ঠিক ততটুকুই পারবে, যতটুকু তার আসে। তার আসার চূড়ান্ত সীমায় তাকে ঠেলে দিয়েও যদি অপারেটিং সিস্টেমের কাজ না-চলে, তখন ভৌত উপাদানগুলোকেই বদলে ফেলার কথা ভাবতে হবে। যেভাবে ভৌত উপাদানের প্রকৌশলগত বদলগুলো ঘটে আর কি।

অনেক মেশিনে এই ভৌত উপাদানদের নিজস্ব জগতের এই নিজস্ব ঘ্যাঁটটা পাকানো হয় সফটওয়ার দিয়ে — তাদের বলে মাইক্রোপ্রোগ্রাম। অন্য অনেক মেশিনে আবার সেই একই কাজটা করা হয় হার্ডওয়ার সার্কিট দিয়ে। এই নিয়ে, কোনো রকম কোনো আলোচনাই আমাদের এই পাঠমালায় আসবে না। উঃ, এটা যে কী রিলিফ, হাফ জানা, কোয়ার্টার জানা, জানি-কি-জানিনা তাই ভালো বুঝছি না — এই সব ঘনচক্কর এলাকা নিয়ে কথা বলে চলাটা তবু সম্ভব, কিন্তু যে এলাকাটা স্পষ্ট জানি যে জানিনা, তাই নিয়ে যদি কথা বলতে হয়?

৩।। অপারেটিং সিস্টেম

যাইহোক, হার্ডওয়ারের যে জটিলতার কথা বলছিলাম। এই জটিলতাটা ফেস করার ফ্রাস্ট্রেশন থেকে পরিত্রাণায় জন্যে ধরায় অবতীর্ণ হয়েছিলেন অপারেটিং সিস্টেমের অবতার। খুব সহজ একটা উদাহরণ নেওয়া যায়। একটা ফ্লপিতে কী কী ফাইল সেটা আমরা জানি কী করে? উইন্ডোজে, এক্সপ্লোরার খুলে, ছবিটা যখন ফুটে উঠল, দুদিকে দুটো পাল্লা — বাঁদিকের পাল্লায় 'এ-ড্রাইভ'-এর গায়ে ক্লিক-মাত্র টাওয়ারের ফ্লপি ড্রাইভের চৌকো দরজাটার গায়ে ছোট্ট আলোটা জ্বলে ওঠে, দু-এক মুহূর্ত পরে ডানদিকের পাল্লায় ফুটে ওঠে সারি সারি ডাইরেক্টরি আর ফাইল। আর গ্নু-লিনাক্সে, এটা গুই হলে, ওই একই রকম ক্লিক করতে হয় '/mnt/floppy'-র গায়ে, একই ভাবে সারি সারি ফোল্ডার আর ফাইল উন্মুক্ত হয়ে যায়। লিনাক্সে অন্য আরো অনেকগুলো সম্ভাবনা থাকে — যাদের নিয়ন্ত্রণ করতে পারেন আপনি নিজেই। হতে পারে যে আপনার ফ্লপিতে মাউন্ট করতে হবে এই ব্যবস্থা আপনি করে রেখেছেন, মাউন্ট করা মানে কী তা আমরা অনেক পরে জানব, অন্তত প্রথম এগারো দিনে কোনো আশা নেই, তখন মাউন্ট করার আগে ক্লিক করলে আপনাকে কিছুই দেখাবে না। হতে পারে যে আপনি কনসোল মোডে কাজ করছেন, তখন আপনি কিবোর্ড থেকে টাইপ করে কমান্ড দিচ্ছেন — '`ls /mnt/floppy`' — মানে, '/mnt/floppy' ডিরেক্টরিতে মাউন্ট করা ফ্লপির ভিতর ফাইল-ডিরেক্টরির তালিকাটা দেখাও। বা, উইনডোজ হলে, মাউস-ক্লিক করার বদলে ডস-প্রম্পট-এ '`dir a:`' কমান্ড দিতে হত। উইনডোজ প্রতিবেশ মানেই মাউস, তা কিন্তু নয়, কনসোল মোডেও কাজ করা যায় বৈকি, এমএসডস (MS-DOS — **M**icro**S**oft-**D**isk-**O**perating-**S**ystem) প্রম্পট বলে একটা জিনিষ দেওয়া থাকে স্টার্ট — প্রোগ্রাম — অ্যাকসেসরিজ মেনুতেই, কিন্তু তাতে লিটারালি খাজনার চেয়ে বাজনা বেশি। কেন, সেটা সম্পূর্ণ অন্য আলোচনা — আমাদের জিএলটি ইশকুলের পাঠমালায় পড়েনা।

এবার, প্রথম ক্লিকের পর খুলে যাওয়া ডিরেক্টরি আর ফাইলের তালিকায় এবার আপনি ফের ক্লিক করেন আপনার পছন্দসই একটা ফাইলের গায়ে। এক মুহূর্তে খুলে যায় ফাইলটা। যদি টেক্সট ফাইল হয় খুলে যায় একটা টেক্সট সফটওয়ার অ্যাপ্লিকেশনের ভিতর। মানে একটা টেক্সট অ্যাপ্লিকেশন খোলে, তার ভিতরে আপনি দেখেন ফাইলটা খোলা আছে, আপনি পড়তে শুরু করেন। আবার যদি ওটা একটা গানের ফাইল হয়, মানে একটা এমপিথ্রি বা ওই গোছের ফাইল, তাহলে আপনি ক্লিক করা মাত্র খুলে যায় একটা গান-শোনার সফটওয়ার এবং তার মধ্যে চলতে শুরু করে আপনার পছন্দসই ওই গানটা। যদি একটা গ্রাফিক্স মানে ছবির ফাইল হয়, আপনি ক্লিক করা মাত্র খুলে যায় একটা ছবি দেখার বা ছবি আঁকার সফটওয়ার এবং তার মধ্যে ফুটে ওঠে আপনার নির্বাচিত ওই ফাইলের ভিতরকার ছবিটা। কিন্তু এটা ঘটছে কী করে?

আপনার ওই ক্লিক করার মুহূর্তটার মধ্যেও অনেক অনুমুহূর্ত থাকে। অপারেটিং সিস্টেমের কাছে আপনি প্রায় ব্রহ্মা। ব্রহ্মার এক আঁখিপল্লবপাতে যেমন ধরাধামে অগণিত যুগ বয়ে যায়, তেমনি আপনার এক মুহূর্তে অপারেটিং সিস্টেম তথা সিপিইউ-র কাছে কেটে যায় কোটি কোটি যুগ, যার মধ্যে অপারেটিং সিস্টেম পরপর করে চলে —

এ-ড্রাইভ বা '/mnt/floppy' নামে আপনি ঠিক কোন জায়গাটাকে বোঝাচ্ছেন সেটা দেখে নেওয়া

আপনি যে নামে আপনার ফাইলটাকে চিহ্নিত করছেন, সেই নামের ফাইলের জন্যে সিস্টেম কোন আইনোড বরাদ্দ রেখেছে সেটা দেখে নেওয়া, এর জন্যে একটা তালিকা আগেই করে রাখা আছে আইনোড টেবল নামে, আপনার পছন্দ করা জায়গাটার সমস্ত ফাইলকে মিলিয়ে (এই যে আইনোড তালিকার কথা বললাম, আইনোড হল একধরনের একটা সংখ্যা-সূচক ব্যবস্থা যা দিয়ে অপারেটিং সিস্টেম একটা ফাইলকে চেনে, আমরা এর সঙ্গে আমরা বিশদ ভাবে পরিচিত হব ছয় এবং সাত নম্বর দিনে গিয়ে)

সেই আইনোড তালিকা থেকেই পড়ে নেওয়া, আপনার নামের ফাইলটার পিছনে আসলে বাস্তব কোন কোন বাইট রয়েছে, তারপর সেই বাইটগুলোয় লিখে রাখা মোট তথ্যটাকে পড়ে নেওয়া

এবার সিস্টেমের ভিতরে আগে থেকেই বানিয়ে রাখা তালিকা থেকে দেখে নেওয়া আপনার দেওয়া ফাইলটা যে ধরনের, সেই ধরনের ফাইল খোলার জন্যে কোন প্রায়োগিক বা অ্যাপ্লিকেশন সফটওয়ার তার চালু করার কথা

এবার আপনার পছন্দ করা জায়গার মানে ডিরেক্টরির পছন্দ করা ফাইলটা থেকে পড়ে নেওয়া গোটা তথ্যটাকে, ওই ধরনের ফাইল খোলার সফটওয়ার চালিয়ে, তার মধ্যে হাজির করা।

জাস্ট ভাবুন আপনার একটিমাত্র নিদেন দুইমাত্র মাউসহেলনের অন্তরালে আপনার অপারেটিং সিস্টেমের মোট কতটা অঘোষিত নির্বাক শ্রম রয়ে যাচ্ছে। এবার, ধরুন এখানে গ্নু-লিনাক্স বা উইন্ডোজ নামের এই অপারেটিং সিস্টেমের নির্বাক মধ্যস্থতাটা নেই, একদম প্রথম দিকের কম্পিউটারে যা হত। সুইচ অন করা থেকেই মেশিনটা সম্পূর্ণ আপনার এবং একমাত্র আপনারই আজ্ঞারই অধীন — আপনিই রাখবেন বা মারবেন, এবং তার চেয়েও ডেনজারাস কথা এই যে কী করে রাখবেন বা কী করে মারবেন তার গোটাটাই আপনাকে জানতে হবে। এই রকম অবস্থায় আপনাকে বলতে হত — ফ্লপি-ড্রাইভের মধ্যের চাকাটা এত ডিগ্রিতে ঘোরাও, এতটা গতিতে, সেখানে প্লাস্টিকের পাতের গায়ে বিদ্যুৎ-আখরে উৎকীর্ণ সংখ্যামালা পড়ো, এই করতে গিয়ে ফ্লপি ড্রাইভের মুণ্ডটা এতটা বেঁকাও ... ইত্যাদি ইত্যাদি।

হার্ডওয়ারের এই জটিলতাগুলোর উপরে একটা আংশিক এবং সাঙ্কেতিক প্রলেপ তৈরি করে দিচ্ছে অপারেটিং সিস্টেম। আংশিক কারণ আপনার সামনে ওই সি-ড্রাইভ, এ-ড্রাইভ, বা '/user/share', '/mnt/floppy' ইত্যাদি সাঙ্কেতিক নামের মাধ্যমে সে তৈরি করে দিচ্ছে একটা ভারচুয়াল বা সাঙ্কেতিক জগত। আপনি ক্লিক করা মাত্র বা '`dir`' বা '`ls`' লিখে এন্টার মারা মাত্র আদত বাস্তব দুনিয়ায় সে কিছু একটা বাস্তব ঘটনা ঘটাচ্ছে যা আপনার কাছে পৌঁছচ্ছে ফাইল-তালিকা হাজির হওয়া নামক সাঙ্কেতিক ঘটনায়।

একটা জিনিষ খেয়াল করে রাখুন, এটা সারা পাঠমালাটা জুড়েই আমরা অনুসরণ করে চলব। যখনই কোনো ইংরিজি বর্ণমালায় লেখা নাম বা শব্দ উল্লেখ করছি আমরা, তার ফন্ট ব্যবহার করছি টাইমস নিউ রোমান, ধরুন '/user/share' বা '/mnt/floppy'। যখন কোনো আদেশ বা কমান্ড, মানে ঠিক যা টাইপ করে কমান্ড প্রম্পটে দিতে হয়, উল্লেখ করছি, যেমন '`ls /user/share`' বা '`mount /mnt/floppy`', তখন ফন্ট ব্যবহার করছি কুরিয়ের নিউ, মানে ফিক্সড উইডথ ফন্ট, যেখানে প্রতিটি অক্ষরই একই পরিমান চওড়া জায়গা নেয়, টাইপরাইটারে যেমন দেখি আমরা। যখন ওই কমান্ডটাকেও আবার নাম হিশেবে আনছি, তখন কিন্তু লিখছি টাইমস নিউ রোমানে। এর সঙ্গে আরো একটা জিনিষ করছি আমরা বাংলার বর্ণমালার সঙ্গে আয়তনগত সাযুয্য রাখার জন্যে টাইমস নিউ রোমান বা কুরিয়ের নিউকে তিন পয়েন্ট কমিয়ে দিচ্ছি। যখন বাংলা ফন্ট ১৪ পয়েন্ট, তখন ইংরিজি ১১, যখন বাংলা ১৩, ইংরিজি ১০, এই রকম।

একটা জিনিষ খুব ভালো করে মাথায় রাখুন, সিস্টেমকে বুঝতে এই ব্যাপারটা বারবার কাজে লাগবে — আপনার এবং অপারেটিং সিস্টেমের এই নাম দেওয়া এবং ব্যবহারের এই গোটাটাই একটা সাঙ্কেতিক ঘটনা। কারণ, আপনার কাছে যা আসছে অক্ষরের নিয়মে সাজানো পরপর বোধগম্য নাম হিসেবে, ধরুন, '/mnt/floppy/song.mp3' যে নাম আপনিই দিয়েছেন, ভৌত অর্থে কিন্তু এই গোটা নামটাই অর্থহীন, হার্ডডিস্কের গায়ে বাইটগুলোর ভৌত শরীরে এরকম কোনো নামই নেই, না 'mnt' বা 'floppy' নামের কোনো ডিরেক্টরি এবং সাবডিরেক্টরি, না 'song.mp3'

নামের কোনো ফাইল। '/mnt/floppy/song.mp3' নামের পিছনে অপারেটিং সিস্টেম যা খুঁজে নিচ্ছে এবং পড়ে তুলে নিচ্ছে নিজের মেমরিতে, যাতে এখুনি কোনো গান চালানোর অ্যাপ্লিকেশন দিয়ে তাকে খুলতে পারে, তারা আসলে কোটি কোটি অর্বুদ অর্বুদ ০ এবং ১ এর শেষহীন সারি। আদত হার্ডডিস্কের শরীরে ওই '/mnt/floppy/song.mp3' নামের কোনো অক্ষরের কোনো অর্থই নেই। কম্পিউটারের কাজের স্বার্থে অপারেটিং সিস্টেম ওই ০ আর ১-এর শেষহীন নিরবচ্ছিন্ন মিছিল থেকে এক একটা ০ আর ১-এর সমাহারের এক একটা সূচক তৈরি করেছে। ফাইল বানানোর ক্রিয়ায় যে সমাহারটা আপনিও বানিয়ে থাকতে পারেন, সিস্টেমও বানিয়ে থাকতে পারে। এই সমাহারগুলোর এক একটা নাম আছে — নামের তালিকা তৈরি হয়েছে, কিছু নাম আপনি দিয়েছেন, কিছু নাম সিস্টেম দিয়েছে, তাও আপনার বোঝার স্বার্থে। এরা হল ফাইল, যাদের এখন থেকে আপনি চিনবেন ওই নাম দিয়ে আর সিস্টেম চিনবে আইনোড তালিকার ওই সূচক দিয়ে।

এবার যখনি আপনি নাম-সর্বস্ব এই সাঙ্কেতিক দুনিয়ায় কোনো একটা নামের উপর কার্সর এনে ইঁদুরের পেট টিপলেন, বা নামটা টাইপ করে কোনো কমান্ড দিলেন, সে ওই সূচক তালিকা থেকে পড়ে নিল — আইনোড — আপনি যা করতে চাইছেন সেটা করল। আপনি ভুলে থাকতে পারলেন ওই সূচক, সূচকের পিছনে সেক্টরের শেষহীন সারি, তাদের মধ্যেকার চৌম্বক বলরেখার সাযুয্য যাকে সিস্টেম ০/১ হয়ে বিট বাইট হয়ে বর্ণমালা বলে চিনছে, ভুলে যেতে পারলেন একটা সেক্টর থেকে অন্য সেক্টরে যেতে আপনার হেডকে, সরি, আপনার না, আপনার সিস্টেমের হার্ড বা ফ্লপি ড্রাইভের হেডকে, কতটা নড়তে হবে, ভুলে যেতে পারলেন অ্যাকচুয়েটর আর মোটর আর তাদের সার্কিট আর তাদের বিদ্যুৎ-বিভবের ওঠাপড়া। আপনি শুধু নাম ধরে পুকারলেন, সে মাখোমাখো হয়ে গেল। কী দুর্ভাগা সেই মহাপুরুষ — শুধু একটা হার্ডডিস্কের বৃহদন্ত্র কখনো ঘাঁটতে হয়নি বলেই বলতে পেরেছিলেন — নামে কী এসে যায়।

শুধু হার্ড ফ্লপি বা কম্পাক্ট ডিস্ক কেন, আপনার প্রিন্টার হলে? তার কাঁটার পিছনে দোয়াতের উদরে কালির হিশেব, আমার প্রেতচুম্বিত ডটম্যাট্রিক্সে, বা আপনার জলচৌকি-পিচকিরি প্রিন্টারের মানে ডেস্কজেটের পিচকিরির — এই গোটাটাই আপনি ভুলে থাকছেন। আপনি 'প্রিন্ট করো' বলেই খালাস, মানে কমান্ড প্রম্পটে কমান্ড দিয়ে, বা, প্রিন্টারের আইকনের গায়ে ক্লিক করে। ওই মুহূর্তটাকেই ভাবুন — আপনি যখন প্রিন্টারের আইকনে ক্লিক করছেন, স্ক্রিনে আরো কিছু ঘটতে শুরু করছে, প্রিন্টের আইকনের গা থেকে একটা করে পাতার ছবি বেরোচ্ছে আর পরমাত্মায় লীন হয়ে যাচ্ছে, মানে প্রিন্ট হচ্ছে — ওই মুহূর্তটাতেই অপারেটিং সিস্টেম আরো কত কী করে চলেছে, আপনার দৃষ্টির অন্তরালে, ভেবে দেখেছেন কখনো? এটা ভাবতে যে হয়না সেটাই অপারেটিং সিস্টেমের সাফল্য।

আপনি মাউসের পেট টিপলেন, মাউসের নাড়িভুড়ির হিশেব রাখতে হল কিন্তু অপারেটিং সিস্টেমকে, স্ক্রিনে ফুটে ওঠা ডিটেইলস-এর সঙ্গে মিলিয়ে সেই মাউস-ইভেন্টকে বুঝল সে। মানে, ভিডিও মনিটরের সমস্ত খুঁটিনাটিও মাথায় রাখতে হয়েছে তাকে। সেই খুঁটিনাটির সঙ্গে মিলিয়ে সে মনিটরে করা আপনার মাউস-ইভেন্ট থেকে মিলিয়ে নিল, কোন ফাইলটাকে চাইছেন আপনি, আপনার ফাইলটা তুলল মেমরির যথাস্থানে। তার জন্যে তাকে মাথায় রাখতে হল মেমরির গঠনের খুঁটিনাটি। এবার মেমরি থেকে প্রিন্টার পোর্টে তাকে পাঠাতে হবে ওই তথ্যটা। তার মানে, কতটা হারে তাকে পাঠাতে হবে আপনার সিস্টেমের বাস বেয়ে সেই হিশেব করল, তার ডিটেইলস, প্রিন্টারের ডিটেইলস, প্রিন্টার পোর্টের ডিটেইলস, তবে পাঠাও তথ্যটা, একগুচ্ছ। আবার পড়ো, আবার স্ক্রিনের ছবি বদলাও, মুহূর্তে মুহূর্তে। অপারেটিং সিস্টেমের মুহূর্ত ঘটে চলেছে আপনার প্রতি মুহূর্তে মাত্র কয়েক কোটি হারে, গোটা সময়টা জুড়েই এই গোটা ঘাপলাটা হ্যান্ডল করছে একা একলা অপারেটিং সিস্টেম। হার্ডওয়ারের ওই খিঁটকেল দুনিয়ার সাপেক্ষে কম্পিউটার জগতের আর সব কিছু — সমস্ত কিছুকে বয়ে নিয়ে চলেছে অপারেটিং সিস্টেম। এর আরো কিছু খুঁটিনাটিতে আমরা আসব পরের দিন, দু নম্বর দিন, তারপরে আসতেই থাকব।

৪।। সিস্টেম প্রোগ্রাম এবং কমান্ড ইন্টারপ্রিটার

আমাদের তালিকাটায় দেখুন, মূল প্রোগ্রাম কাঠামো বা সিস্টেম প্রোগ্রামের এলাকা শুরু হয় অপারেটিং সিস্টেম থেকে। একদম নিচে হার্ডওয়ার থেকে গুনলে এটা ৪ নম্বর স্তর। এর উপরে মানে সিস্টেম প্রোগ্রামিং-এ এর উপরের মানে ৫ নম্বর স্তরে আছে তিনটে উপাদান — কম্পাইলার, এডিটর, কমান্ড ইন্টারপ্রিটার। অপারেটিং সিস্টেম এবং

তার ঠিক উপরের স্তরেই এই তিনটেকে মিলিয়ে তৈরি সিস্টেম প্রোগ্রামের এলাকা। কমান্ড ইন্টারপ্রিটার বলতে সেই সফটওয়ার যা আমাদের কমান্ডকে ইন্টারপ্রিট করে মেশিনের কাছে পৌঁছে দেয় এবং তাকে ওই কমান্ড পালন করায়। এই সংজ্ঞাটা আমায় শ্রীপঞ্চমীর পুণ্য প্রভাতে প্রণামনীয় দ্বিতীয় ঠাকুরের একটা গপ্পো মনে পড়াল। দর্শনের এমএ ক্লাসে এক মাস্টারমশাই, সে ব্যাটা ইংরেজ, নইলে এতাবৎ কলোনাইজার কনফিডেন্স আসবে কোত্থেকে, 'বিড়াল' নামক প্রাণীর পরিচয় দিতে গিয়ে বলে দিচ্ছে, 'ক্যাট' মানে হল 'এ ফেলাইন কোয়াড্রাপেড'। তবু যে ব্যাটা সাপ্লিমেন্ট অফ কপুলা দিয়ে বলেনি এটা আমাদের কলেনাইজডদের বাপের ভাগ্য। ওই সাহেবসুবোর স্টেটমেন্টটা খেয়াল করুন, আমাদের বিদ্যাচর্চার বিপুল সংখ্যক সংজ্ঞা ঠিক ওই বর্গের। 'মানে' ইত্যাদি দ্বিতীয় স্টেটমেন্টটাই যেখানে মূল মানেটাকে আরো ঘেঁটে দিচ্ছে, সংজ্ঞাটা শুনে যেখানে গুলিয়ে যাচ্ছে আরো, বিড়াল যে কাকে বলে সেটাই আপনি ভুলে যাচ্ছেন। কমান্ড ইন্টারপ্রিটারের সংজ্ঞাটা শুনে ঠিক তাই হল না? এবার একটা উদাহরণ দিয়ে বোঝা যাক।

ধরুন আমরা আগে যে উদাহরণটা দিলাম, '`ls /mnt/floppy`' বা '`dir a:`' — এখানে 'ls' বা 'dir' বা বলায় কম্পিউটারের অপারেটিং সিস্টেমকে যে ফ্লপি ড্রাইভে ঢোকানো ফ্লপিডিস্কের শরীরে লেখা ফাইল ডাইরেক্টরির তালিকাটাই হাজির করতে হবে স্ক্রিনে এটা অপারেটিং সিস্টেমকে বুঝিয়ে দিচ্ছে এই কমান্ড ইন্টারপ্রিটার। আসলে এখানে জটিল একটা প্রক্রিয়া শুরু হয় যা আমরা ঠিক করে বুঝতে শুরু করব পাঁচ নম্বর দিন থেকে। মোদ্দা ব্যাপারটা অনেকটা এইরকম, ব্যবহারকারীর সঙ্গে সরাসরি চারচক্ষুর মিলন হচ্ছে এই কমান্ড ইন্টারপ্রিটার-এর। কমান্ড ইন্টারপ্রিটার এই টাইপ করে দেওয়া পুরো জিনিষটা মানে '`ls /mnt/floppy`' শব্দগুলো থেকে 'ls' অংশটাকে আলাদা করে ফেলে। এই 'ls' টুকু হল আদত কমান্ড। অন্যটুকু মানে '/mnt/floppy' হল তার আর্গুমেন্ট বা যার উপর সে কমান্ডটা চালাবে। এর সঙ্গে আরো একটা অংশ থাকতে পারত, যার নাম 'অপশান' বা পছন্দ। ভালো করে আলোচনাটা ধরব আমরা ছয় নম্বর দিনের গোড়ায় গিয়ে।

ধরুন কমান্ডটা যদি হত '`ls -al /mnt/floppy`', তখন ওই হাইফেন বা '-'-এর পরের '`al`' অংশটা হত অপশান বা পছন্দ। এতে জানানো হত যে, ফ্লপিতে যদি এমন কোনো ফাইল থাকে যা সিস্টেম স্বাভাবিকভাবে দেখায় না, যেমন সিস্টেম ফাইল, সেটাকেও দেখাও। এবং দেখাও তাদের সম্পর্কিত সব খুঁটিনাটি, কে সেই ফাইল-এর মালিক, কে কে ফাইলটা পড়তে বা লিখতে বা চালাতে পারে, কখন ফাইলটা বানানো হয়েছিল, ফাইলের সাইজ, ইত্যাদি সমস্ত খুঁটিনাটি। এখানে 'al'-এর 'a' মানে অল, বোঝাচ্ছে সব ফাইল দেখাও, আর 'l' মানে লং, দীর্ঘ বিবরণের আকারে সমস্ত খুঁটিনাটি দেখাও ফাইলের। উইনডোজ প্রতিবেশে কমান্ডটা হত '`dir/a a:`' — দীর্ঘ বিবরণ দিতে বলার সুযোগটা উইনডোজে পাওয়ার ব্যবস্থা থাকেনা। যে অধিকার এবং অনুমতির নিরাপত্তাব্যবস্থার খুঁটিনাটি হিশেবে ওই উপাদানগুলো থাকে বা আসে, ফাইলসিস্টেমের গঠনের মধ্যে গেঁথে দেওয়া সেই নিরাপত্তাব্যবস্থাটাই নেই উইনডোজে। সাত নম্বর দিনে এই বিষয়ে আসব আমরা, কেন গ্নু-লিনাক্স একটা কাঠামোর ভাইরাস আক্রান্ত হওয়াটা সংজ্ঞাগত ভাবেই অসম্ভব। যাইহোক, ফিরে আসা যাক কমান্ড ইন্টারপ্রিটার-এর কথায়। অপশন সহ এই গোটা কমান্ডটা পালন করবে শেষ অব্দি অপারেটিং সিস্টেম, কিন্তু তার আগে এই গোটা কমান্ডটা কোথায় আছে, তার অপশানগুলোর ঠিক কেজো অর্থটা ঠিক কী দাঁড়াচ্ছে — এই গোটাটা আপনার কমান্ড পাওয়ার পর অপারেটিং সিস্টেমকে বুঝিয়ে দেওয়ার দায়িত্ব কমান্ড ইন্টারপ্রিটারের। পাঁচ নম্বর দিনে কিছুটা এবং তার পরে দশ নম্বর দিনের গোটাটা জুড়েই এই কমান্ড ইন্টারপ্রিটারের ব্যাকরণ শিখব আমরা। ভেবে দেখুন, আপনার কমান্ড হল এমন একটা কিছু যা কাজ করে, মানে, চলে। অর্থাৎ প্রোগ্রাম। মানে এমন একটা ফাইল যা চলে, আগেই যা বললাম। তার মানে এই ফাইলটা এমন কোথাও থাকতে হবে যেখান থেকে তাকে খুঁজে পেতে পারে কমান্ড ইন্টারপ্রিটার, এবং সেই কর্তব্যকর্মটা, তার সমস্ত শর্ত এবং বাধ্যতাসহ, তখন অপারেটিং সিস্টেমের হাতে হস্তান্তর করতে পারে। কোথায় কোথায় ফাইলটাকে খুঁজে দেখবে কমান্ড ইন্টারপ্রিটার বা শেল, গ্নু-লিনাক্সে সবচেয়ে জনপ্রিয় শেল ব্যাশ, সেই হদিশটা থাকে পথনির্দেশ বা 'PATH' নামক সিস্টেম ভ্যারিয়েবলে। এর কথাও পরে জানব আমরা।

মানে, মোদ্দা কথা হল, কমান্ড দিলে সেই কমান্ডটা তো একটা শব্দ এবং অক্ষরের সমাহার — সেই সমাহারের অর্থোদ্ধার করার দায়িত্ব কমান্ড ইন্টারপ্রিটার বা শেলের। ইনফ্যাক্ট, এবার অর্থ মোতাবেক কাজ করার দায়িত্ব অপারেটিং সিস্টেমকে ন্যস্ত করে কমান্ড ইন্টারপ্রিটার এবার ঘুমোতে চলে যায়। ছবি টবি দিয়ে এর একটা সুন্দর আলোচনা আছে পাঁচ নম্বর দিনে। তবে, সবসময় এই শেলকে খুব আলাদা করে খেয়াল করা যায়না। বিশেষ করে

যখন গুই বা গ্রাফিকাল ইউজার ইন্টারফেস-এ কাজ করছি, উইনডোজ যেমন, বা গ্নু-লিনাক্স-এ কেডিই বা গুহনোম বা ফ্লাক্সবক্স বা ব্লাকবক্স বা এনলাইটেনমেন্ট বা এক্সএফসিই ইত্যাদি, সেখানে কিন্তু সামনা সামনি প্রকাশ্যে এই কমান্ড ইন্টারপ্রিটারটাকে প্রায়ই ধরা যায়না। বোঝা যায়, ব্যাশে নেমে কী কী প্রসেস বা প্রক্রিয়া তখন সিস্টেমে চালু রয়েছে সেটা দেখতে চাইলে, 'ps aux' জাতীয় কমান্ড দিয়ে। দেখা যায় চলতে থাকা সবকটা প্রোগ্রামেরই উল্লেখ আছে তাতে। এবং তাদের সবারই মা-প্রক্রিয়া হল ব্যাশ, সেটাও মিলিয়ে নেওয়া, কী করে — সেটা আমরা পরে দেখব। এই প্রক্রিয়া বা প্রসেস মানে একটা প্রোগ্রামের একবার চলা, এর বাস্তব অর্থটা কী দাঁড়ায় সেটা আমরা ধীরে ধীরে বুঝব।

ধরুন 'move' বা 'mv' বা কমান্ডটাকেই, উইন্ডোজ-এর বেলায় 'move', আর লিনাক্স-এর বেলায় 'mv'। উইন্ডোজ-এ যদি আমরা মাউস ক্লিক করে একটা ফাইলকে অন্য জায়গায় নড়াই সেখানেও সামনের ওই ফোল্ডার থেকে ফোল্ডারে ফাইল উড়ে যাওয়ার ছবির পিছনে আসলে কাজ করে এই কমান্ডটাই, আমাদের অগোচরে। আর প্রত্যক্ষে এটাকে আমরা দেখতে পাই যখন এমএস-ডস প্রম্পট খুলে সরাসরি কমান্ড ইন্টারপ্রিটার দিয়ে কাজ করি। আমরা মাউস টিপে বা কিবোর্ড থেকে ওই কমান্ড-টা দিলে কী করতে হবে সেটা মেশিনকে জানায় কমান্ড ইন্টারপ্রিটার। যেমন উইন্ডোজ-এর বেলায় এই কমান্ড ইন্টারপ্রিটারের নাম 'command.com'। এই ফাইলটা থাকে উইন্ডোজ-এর রুট ডিরেক্টরিতে, মানে স্বাভাবিক একটা সিস্টেমের বেলায় C:\ -এ। শুধু এই ফাইলটা নয় অবশ্য, কমান্ড ইন্টারপ্রিটার-এর গোটা আবহটা উইনডোজ প্রতিবেশে বানিয়ে তোলে আরো কিছু ফাইল, যেমন 'msdos.sys' আর 'io.sys'। এরাও থাকে ওই রুট ডিরেক্টরিতেই। তবে পরের দুটো গোপন ফাইল, দেখতে চাইলে কমান্ড প্রম্পটে দিতে হয় 'dir/a c:' বা গুই মোডে থাকলে এক্সপ্লোরারের ভিউ মেনুতে 'শো অল ফাইলস' অপশান অন করতে হয়। আর লিনাক্স-এর বেলায় এই কাজটা করে শেল, মানে সচরাচর যার নাম bash — যদিও এছাড়াও অন্য শেল আছে। তিনি থাকেন রুট ডিরেক্টরির ভিতর '/bin' ডিরেক্টরিতে, তার ঠিকানাটা হল '/bin/bash'। এই ঠিকানার ব্যাপারটাও ঠিক ভাবে বুঝতে আমাদের ছয় নম্বর দিন অব্দি ওয়েট করতে হবে।

৫।। এডিটর

দ্বিতীয়টা হল এডিটর, যাকে বলা যায় সবচেয়ে সরল সাদামাঠা ওয়ার্ড প্রসেসর। দিন শূন্যয় আমরা যাকে বিশুদ্ধ টেক্সট-এর বিশুদ্ধ এডিটর বলে ডেকেছিলাম। ওয়ার্ড প্রসেসর বলতে আমাদের মাথায় থাকে ওপেন সোর্স জগতের ওপেন অফিস অর্গ রাইটার বা মাইক্রোসফট-এর ওয়ার্ড বা লোটাস-এর স্মার্ট স্যুট বা সিমানটেক-এর পেজমেকার। কিন্তু ভিতরে ভিতরে এর প্রত্যেকটাই হল একটা সরল সাদামাঠা এডিটর যা অক্ষরের সমাহারকে হাজির করে, এবং সেই সরল সাদামাটা এডিটরের সঙ্গে বাড়তি আরো কিছু ফর্মাটিং — অক্ষরগুলোর কোনো বিশেষ আয়তন বা রং বা বিশেষ কোনো চেহারা বা ফন্ট বা অন্য কিছু যেমন ব্লক বা ইটালিক বা আন্ডারলাইন ইত্যাদি আছে কিনা, প্যারাগ্রাফগুলোর কোনো ফর্মাটিং আছে কিনা, , বা কোনো সারণী বা ছবি বা কোনো বিশেষ কিছু আছে কিনা, ইত্যাদি। তাদের ঠিক কী ভাবে সাজানো আছে পাতায়, মানে তাদের লে-আউট ইত্যাদি।

এই গোটা গল্পটাকে বুঝতে পারবেন বেশ ভালো করে, হাতেকলমে, যদি একটা ওয়ার্ডপ্রসেসর ফাইলকে কমান্ড এডিটর দিয়ে খোলেন। কমান্ড এডিটর নামটা কেন তাতো আগেই বলেছি, এক বা একাধিক কমান্ডের তালিকাকে লিখে ফেলা যায় এবং এডিট করা যায় এই কমান্ড এডিটর দিয়ে, কোনো ধরনের কোনো ফর্মাটিং-এর ধার ধারেনা, বিশুদ্ধ নিখাদ আছোলা টেক্সট। ঠিক যেমন কম্পিউটারের অপারেটিং সিস্টেম যেমন কোনো কমান্ড পালন করতে গিয়ে বাড়তি কোনো ফর্মাটিং পায় তার ভিতর, মাঝখান থেকে কমান্ড বা আদেশটাই ধরতে পারবে না, এবং তাই পালন করতে পারবে না। এই পরপর কমান্ডগুলোকে একসঙ্গে এনে যে ফাইলগুলো আমরা তৈরি করি কমান্ড এডিটর দিয়ে, তাদের চালু নাম স্ক্রিপ্ট । এই পাঠমালার শেষ মানে দশ নম্বর দিনটা আমরা গোটাটাই প্রায় ব্যয় করব ব্যাশ শেলকে কাজ করানোর নানা ধরনের এই স্ক্রিপ্ট দিয়ে। এই স্ক্রিপ্টগুলো গ্নু-লিনাক্স মেশিনে ভারি গুরুত্বপূর্ণ একটা ব্যাপার। এর মধ্যে থাকে পরপর যে কমান্ডগুলো আপনি কম্পিউটারকে দিয়ে করাতে চান। কম্পিউটার যে কমান্ড পাবে সেখানে তো ফর্মাটিং থাকলে চলবে না। ফর্মাটিং মানে তো মূল বিশুদ্ধ টেক্সট যোগ আরো কিছু চিহ্ন। যে চিহ্নগুলো ওই ফর্মাটিংকে হাজির করবেন। যে কোনো ওয়ার্ডপ্রসেসরের মধ্যে 'শো ননপ্রিন্টিং ক্যারেকটারস'

অপশানটা অন করে দিলে পাতা জুড়ে লেখার সঙ্গে সঙ্গে যে চিত্তির-বিচিত্তিরগুলো ফুটে ওঠে। সেই চিহ্নগুলো মূল কমান্ডকেই আটকে দেবে, কম্পিউটারের কাছে সঠিক কমান্ড পৌঁছেই দেওয়া যাবেনা।

এটা একদম হাড়ে হাড়ে বুঝবেন, ধরুন আপনি ওয়ার্ডপ্রসেসর ওপন-অফিসে লেখা একটা ফাইল যদি কমান্ড এডিটর ইম্যাক্স বা ভিম দিয়ে খোলেন, বা এমএসওয়ার্ডে লেখা ফাইল যদি নোটপ্যাড দিয়ে খোলেন। মূল টেক্সট-এর একটা জায়গা চিনে রাখুন — তারপর সার্চ দিন সেই অক্ষরমালা দিয়ে, টেকনিকাল ভাষায় যার নাম স্ট্রিং, দেখবেন, সবটাই আছে, পুরো টেক্সটাই, শুধু কিছু অক্ষর কিছু চিহ্ন একটু অন্য চেহারায় দেখতে পারেন, যাদের যে চেহারায় দেখানোর উপায় নেই ওই সাদামাঠা কমান্ড এডিটরের। এবং পাবেন আরো বহু চিহ্ন এবং সংখ্যা, প্রচুর, যাদের কোনো অর্থ করতে পারবেন না।

এই সরল আকারে এডিটরকে পেতে লিনাক্স জগতে আছে প্রচুর কিছু, যার মধ্যে প্রমুখতম হল ইম্যাক্স এবং ভিম। এবং উইন্ডোজ জগতে আছে এডিট এবং নোটপ্যাড। এর মধ্যে এডিট বস্তুটা ডস-এর সময় থেকেই আছে — এখনো ২০০০ এবং এক্সপিতেও দেওয়া থাকে। ডস প্রম্পট-এ সরাসরি '`edit textfile`' লিখে এন্টার মারলে দেখবেন একটা আদিম কালের চাঁদিম হিম গোছের কালচে ধূসর উইন্ডো খুলে গেল, তার মধ্যে একটা ফাইল খোলা রয়েছে, যার নাম 'textfile' এবং সেখানে এবার আপনি যা টাইপ করবেন সেটাই ফুটে উঠবে এই ফাইলে। শেষে আপনি সেভ করে বেরোবেন।

এই সাদামাটা কমান্ড এডিটরগুলো প্রাথমিক ভাবে যা কাজ করে তা হল অনেক কমান্ডকে একসঙ্গে জুড়ে একত্রে হাজির করা। কোনো ফর্মাটিং কোনো আড়ম্বর কোনো কারুশিল্প ব্যতিরেকে। কারণ ওই কারুশিল্পের প্রয়োজন পড়ে আমাদের, কম্পিউটারের নয়। যেমন উইন্ডোজ-এর ব্যাচ ফাইল লেখা হয়। উইন্ডোজ ইনস্টলেশন-এ C:\ বা রুট-ডিরেক্টরিতে যেমন আমরা ব্যবহার করতাম, খুব কাজে লাগত ফাইলটা, 'autoexec.bat'। এছাড়াও নানা কাজে ছোটখাটো নানা ব্যাচ ফাইল ব্যবহার করেছি। আমার মত বুড়ো কেউ যদি এই লেখাটা পড়ে, তারই বোধহয় এটা পরিচিত লাগতে পারে, ডস-প্রম্পটে ওয়ার্ডস্টার দিয়ে টাইপরাইটার হিশেবে কম্পিউটার ব্যবহার করা দিয়ে যাদের কম্পিউটারের সঙ্গে ফোরপ্লে শুরু হয়েছিল। এত আইডেন্টিফাই করেছিলাম রোমান পোলানস্কির বিটার মুন সিনেমার ওই বুড়ো লেখকের সঙ্গে — ব্যাটা ওয়ার্ডস্টারে লিখছে। তখন নানা কাজে লাগত ব্যাচ ফাইল।

এই ব্যাচ ফাইলের পরপর লেখা কমান্ডগুলোকে একের পর এক পালন করতে থাকে কম্পিউটার। একবার ভাবুন তো — কম্পিউটার প্রোগ্রাম ব্যাপারটা আসলে কী? পরপর করে যাওয়ার দীর্ঘ আদেশ তালিকা ছাড়া? সেদিক থেকে এই ব্যাচ ফাইলকে সি বা ফোর্ট্রান বা সিপ্লাসপ্লাস বা জাভা বা পার্ল বা পাইথন বা অন্য যে কোনো কম্পিউটার ল্যাংগুয়েজ-এ লেখা প্রোগ্রামের ডিম ভাবা যেতে পারে যা ফুটে এখনো ছানা বেরোয় নি। খুব বিশদ করে এই প্রসঙ্গটায় আমরা আসব পাঠমালার শেষ দিনে।

এবার ধরুন, এই কমান্ড এডিটর, বা নির্দেশ-সম্পাদক, যে নামেই আপনি ডাকুন, তাই দিয়ে আপনি একটা প্রোগ্রাম লিখলেন। আট নয় বা দশ নম্বর দিনে এরকম শেল স্ক্রিপ্ট আমাদের লাটকে লাট পড়তে হবে। তখন দেখবেন, যেটুকু কম্পিউটার ভাষা জানলে ওটা বোঝা যায় সেটুকু জানা থাকলে তো বুঝতেই পারবেন গোটাটা, আর না-থাকলেও অন্তত কিছু বুঝতে পারবেন। মানে সেটা একটা বোধ্য জগত। বিশেষ কিছু শব্দের বা ব্যবহারের বিশেষ অর্থগুলো জানা থাকলেই যার মানে বোঝা যায়। যেমন ধরুন একটা শক্ত এবং অপরিচিত বিষয়ের বই খুললে আমাদের হয়, বুঝতে পারি যে এটা বোঝার যোগ্য, শুধু বিষয়টা জানিনা বলে বুঝতে পারছি না। এর সঙ্গে কিন্তু একটা বিরাট ফারাক আছে আমাদের পরিচিত প্রোগ্রামগুলোর, যা দিয়ে আমরা লেখালেখির বা গানশোনার বা ছবি-আঁকার অ্যাপ্লিকেশন চালাই। ধরুন, গ্নু-লিনাক্স জগতের অ্যাবি-ওয়ার্ড আর এমএস-উইনডোজের এমএস-ওয়ার্ড, এরা দুজনেই দুটো প্রোগ্রাম, মানে, এমন ফাইল যে চলে। আমরা যখনি ওয়ার্ড-প্রসেসর চালাই, তখন আমাদের ক্লিক করা বা কিবোর্ডে কমান্ড দেওয়া মাত্র অপারেটিং সিস্টেম আসলে এই প্রোগ্রামগুলোকে চালায়। এই রকম কোনো ওয়ার্ডপ্রসেসর, বা গ্রাফিক্স প্রোগ্রাম, বা অন্য কোনো প্রোগ্রামকে সরল সাদামাটা কমান্ড এডিটর দিয়ে খুলুন তো? কী পেলেন, পড়ে কী বুঝছেন? কী দেখছেন? ঠিক আমাদের শূন্য নম্বর দিনের ভিশুয়াল চ্যাংড়ামোর মত একটা স্ক্রিন তো? তাহলে? এই তফাতটা কেন?

এটা বুঝতে গেলে আমাদের কম্পাইলার ব্যাপারটা কী সেটা জানতে হবে। কম্পাইলার, আগেই বলেছি, একটা দোভাষি, যা মানববোধ্য ভাষাকে কম্পিউটার বোধ্য করে তোলে। অনুবাদ করে যেমন আমরা অন্য ভাষার গল্প পড়ি। মানে বুঝতে পারি। এই কম্পাইলার হল আমাদের একটু আগে করা টেবিলে সিস্টেম প্রোগ্রাম স্তরে আমাদের তালিকার শেষ বস্তুটা।

৬।। কম্পাইলার

একটা খুব ছোট্ট সি প্রোগ্রাম ভাবা যাক। ধরুন তার নাম দিয়েছেন 'gadha.c' — একে বলে সি-এর কোড ফাইল। লেখা হয়েছে কোনো একটা কমান্ড এডিটর দিয়ে। এটা মোটামুটি সর্বজনগ্রাহ্য একটা প্রথা, সি নামের প্রোগ্রামিং ভাষায় লেখা কোড মানে প্রোগ্রাম ফাইলের নাম শেষ হবে, '.c' দিয়ে। সুগারকে শুগার বা লিজেন্ড-কে লেজেন্ড বা ডেভেলপমেন্ট-কে ডিভলেপমন্ট বলার মত, ধাপে ধাপে দামীতর বকলসের কুকুর হওয়ার প্রক্রিয়ায়, নিউব্যারাকপুর কিশলয় ইশকুলের 'টাইটেল' শব্দটা দিব্য বদলে গেছিল কলেজস্ট্রিটের 'সারনেম' বা 'পদবি' বা আরো পরে 'লাস্টনেম' শব্দে, রীতিমত লজ্জা পেতাম একসময়, ভুল বলে ফেললাম না তো, অ্যান্ড নাউ দ্যাট আই অ্যাম নাইন্টিথ্রি, ডোন্ট কেয়ার এ ড্যাম ইউ সি, তাই টাইটেলেই চলুক। ফাইলনামের এই '.c' টাইটেল দেখলেই আপনি বা আপনার সিস্টেম দুজনেই চিনে নিতে পারবেন এটায় ভরা আছে সি ভাষার কোড। এবার এই 'gadha.c' ফাইলের কোডটা দেখুন।

```
#include <stdio.h>
int main(void)
{
     printf("\nKire Gadha!!!\n");
     return 0;
}
```

এই প্রোগ্রামটা আর কিছুই না — কম্পিউটারকে বলে উপরে আর নিচে একটা করে ফাঁকা লাইন রেখে স্ক্রিনে ইংরিজি অক্ষরে ফুটিয়ে তোলো — 'কিরে গাধা!!!'। যার 'কিরে গাধা!!!' ফুটিয়ে তুলতে এতগুলো লাইন নির্দেশ লাগে, তাকে গাধা ছাড়া কী বলবেন? এই লাইনগুলোয় মানে 'gadha.c' প্রোগ্রামটায় কী আছে, তার একটা আলগা আন্দাজ নেওয়ার চেষ্টা করা যাক।

> উপরের দুটো লাইন, '#include <stdio.h>', 'int main(void)' এবং '{', আর একদম শেষের লাইন, '}' — এই চারটে লাইন অনেকটা সি-সঙ্গীতের ধ্রুবপদের মত, যা কেউ শোনেনা, গানের মাঝে ওইখানটায় নামিয়ে নেয় পেন্ডিং হাঁচিটা। এবং পাঁচ নম্বর লাইনও, মানে 'return 0;', আদতে সেই একই, যে ধ্রুবপদ দিয়েছো বাঁধি। কিন্তু মজার কথা কী বলুন তো, সেটা বোঝার মত বুদ্ধিটুকুও আপনার মেশিনের নেই, তাকে ওই লাইনগুলো দিয়ে দিতেই হবে। নইলে একটা দায়িত্বশীল নাগরিকের ভোটারলিস্টেই আপনি প্রোগ্রামটাকে তুলতে পারবেন না, ভোট দেওয়া তো দূরের কথা। এর উল্টোদিকে আবার, কীর্তনের মূল লাইনেরও কোনো অর্থ নেই মালের কাছে। এখানে 'কিরে গাধা' না বলে 'কিগো সোনা' বললেও তার জ্যাবড়া মুখের একটা রেখাও বদলাত না। দাঁড়ান দাঁড়ান, কম্পিউটারের মুখ কোনটা — এটা কোনোদিন ভাবিনি তো?
>
> দোহারকির প্রথম লাইন, মানে '#include <stdio.h>', জানায়, সি-এর নিজস্ব সঙ্কেতে, যে, এই প্রোগ্রামে আমরা সি-এর স্বাভাবিক ইনপুট আউটপুট ফাংশনকে ব্যবহার করব — স্ট্যান্ডার্ড ইনপুট আউটপুট — স্ট্যান্ডার্ড-এর এসটিডি (std) এবং ইনপুটআউটপুট-এর আইও (io)। আর ডটএইচ (.h) হল সেই সাফিক্স যা সি-র সমস্ত এই জাতীয় ফাইলের শেষে থাকে, হেডার ফাইল — যা থেকে এদের চেনা যায়। কেন এটাকে দোহারকি বলেছিলাম জানেন, ধরুন, গান গেয়ে ওঠার গায়ক যদি বলে, বন্ধুগণ, এবার আমি আবার গলা ব্যবহার করে আপনাদের গান শোনাতে যাচ্ছি। অ্যাজ-ইফ, খুব চেষ্টা করলে সে তার কান বা পা ব্যবহার করেও গাইতে পারত।
>
> দ্বিতীয় লাইনটা, মানে 'int main(void)', আমাদের জানায় যে, 'gadha.c' নামের এই কুচো প্রোগ্রামটায় আমরা যে ফাংশনটা গড়ে তুলব তা আমাদের শব্দ ফুটিয়ে তোলার কাজটা করে দেবে কিন্তু তার ফাংশন হিশেবে চরিত্রটা পূর্ণসংখ্যার মানের, মানে ইন্টিজার, যার চিহ্ন ইন্ট (int)' এবং এই প্রোগ্রামটার জন্যে তার

নিজের কোনো ভ্যালু বা মান থাকবে না। অর্থাৎ, সেটা শূন্যগর্ভ বা ভয়েড (void)। আর এই ফাংশনটা মূল বা মেইন (main) ফাংশন এই অর্থে যে তার সঙ্গে আরো চামচা ফাংশন থাকতেই পারত, অন্য অনেক প্রোগ্রামে তা থাকেও। এই কুচো প্রোগ্রামটার বেলায় যা নেই। এই ফাংশন ব্যাপারটা সি-ভাষার ব্যাকরণ মোতাবেক বাধ্যতামূলক, সেই জায়গাটুকু বানিয়ে দেয় — যেখানে দাঁড়িয়ে সি গোটা কাজটা করে। যাকগে, সেই আলোচনার জায়গা এটা নয়।

তিন এবং ছয় নম্বর লাইন মানে শুরু আর শেষের ব্রেস বা সেকেন্ড ব্রাকেট মানে, '{' আর '}' — এরা দুজনে মিলে দেখায় ওই মেইন ফাংশনের শুরু আর শেষ।

পাঁচ নম্বর লাইন, মানে 'return 0;', ওই শূন্যগর্ভ ফাংশনের শূন্যগর্ভতাকেই আর একবার ঘোষণা করে। জানায় যে, সত্যিই ফাংশনটা একটা শূন্য ভ্যালু ফিরিয়ে দিয়েছে। আসলে এটা একটা সিগনাল, ইউনিক্স এবং সি-তে প্রবল ভাবে ব্যবহৃত সিগনালটা, এর অর্থ এই যে, কাজটা সফল ভাবে শেষ হয়েছে। লাইনটার শেষে যে যতিচিহ্নটা, ';', এটা সি নামক কম্পিউটার ভাষার ব্যাকরণ অনুযায়ী কোনো লাইনের বাধ্যতামূলক যতিচিহ্ন। দেখুন দুটো সেকেন্ড ব্রাকেটের মধ্যে থাকা প্রোগ্রামের যে মূল অংশ, তার দুটো লাইনই শেষ হয়েছে এই যতিচিহ্নে।

একমাত্র পড়ে রইল 'gadha.c' প্রোগ্রামের চার নম্বর লাইন 'printf("\nKire Gadha!!!\n");'। আদতে প্রোগ্রাম মানে এটুকুই। এরও শেষে সেই বাধ্যতামূলক যতিচিহ্ন ';'। সেটাকে বাদ দিলে যেটুকু রইল তার মধ্যে একটা হল সি ভাষার একটা নির্দেশ, প্রিন্ট ফাংশন 'printf'। প্রিন্টএফ ফাংশন দিয়ে জানানো হচ্ছে, তার পরের ব্রাকেটের মধ্যে, দুই ডাবল-কোট বা '"' চিহ্নের মধ্যের গোটাটা স্ক্রিনে প্রিন্ট করতে মানে ফুটিয়ে তুলতে। মধ্যের অংশটা খেয়াল করুন, '\nKire Gadha!!!\n'। এই অংশটার শুরু আর শেষে রয়েছে দুই '\n', যার মানে, আমরা বোধহয় আন্দাজ করতে পারছি, নিউলাইন বা নতুন-লাইন। মানে দুবার জানানো হচ্ছে এক লাইন করে ছাড়তে, একবার ফুটিয়ে তোলা কথাটার শুরুতে, আর একবার শেষে। আর এই দুই নিউলাইনের মধ্যে রয়েছে ফুটিয়ে তোলার বাক্যাংশটা, আমাদের সেই সম্ভাষণ, 'Kire Gadha!!!'। এই দুর্ধর্ষ রকমের সমাদরপ্রবণ লাইনটাকে আপনি যাতে উপরে এবং নিচে দুটো ফাঁকা মানে কালো লাইনের প্রেক্ষাপটে তার সমূহ স্পষ্টতায় স্ক্রিনের কালো শরীরে ফুটে উঠতে দেখতে পারেন।

আপনারা যারা একটুও সি জানেন না তাদের কাছেও নিশ্চয়ই আর জটিল লাগছে না। ম্যাক্সিমাম একটু অপরিচিত লাগতে পারে গোটা ব্যাপারটা, কিন্তু মোদ্দা ব্যাপারটা খায় না মাথায় দেয় সেটা বেশ ভালোই বোঝা যাচ্ছে। কিন্তু মজার কথা কী বলুন তো? এই এটুকুও কিন্তু একটা কম্পিউটারের কাছে দুর্বোধ্য নয় শুধু, সম্পূর্ণ অবোধ্য। এই অবোধ্য ব্যাপারটাকে কম্পিউটারের কাছে বোধ্য তাই পালনীয় করে তোলে কম্পাইলার।

বলেছি আগেই, কম্পিউটার শুধু বোঝে ০ আর ১। একটা সার্কিটে হয় বিদ্যুৎ আছে, অথবা নেই। এবং এর সুবঘীর বোধ এবং জ্ঞানভাণ্ডারের এখানেই শেষ নয়। শুধু ০ আর ১ বোঝাই নয়, আমাদের কম্পু-বাবু এমনকি তাদের যোগ-ও করতে পারেন। ০ আর ০ মানে শূন্য, এবং ০ আর ১ মানে ১। কম্পিউটার যাই করুক, সার্চ ফর এক্সট্রাটেরেস্ট্রিয়াল ইন্টেলিজেন্স থেকে হিউম্যান জিনোম, সবই এই ০ আর ১ আর যোগ। এবার গুণ মানে অনেক যোগ, বিয়োগ মানে যোগের উল্টো। আর ভাগ মানে গুণের উল্টো। এভাবেই গোটাটা। তাই কম্পিউটার এমনই গাধা যে আমাদের ওই গাধোচিত প্রোগ্রামটিকেও সরাসরি দিলে কম্পিউটার কিছুই বুঝতে পারেনা।

সি বা যে কোনো কম্পিউটার ভাষাই, সেই অর্থে তাই, কম্পিউটারের ভাষা নয়, কম্পিউটার চালকের ভাষা, প্রোগ্রামচির ভাষা, কী করিতে হইবে এটা কম্পিউটারকে যে বলে দেয় সেই প্রোগ্রামের রচয়িতার ভাষা। কম্পিউটার কাজ করে নিতান্তই ওই একটু আগে দেখানো টেবিলের ১ আর ২ নম্বর স্তর মানে ভৌত উপাদান আর মাইক্রোআর্কিটেকচার দিয়ে। সেই জগতের উপর চলে ওই ০ আর ১ আর +, মানে ০ আর ১ দিয়ে লেখা তথ্য এবং তাদের যোগ করে চটকানো। সব জটিলতার তথ্যই আসলে তৈরি এই সরলতম ০ আর ১ দিয়ে। এবং সব জটিলতার সব চটকানোই তৈরি এই যোগক্রিয়ার নানা রকম সমাহার দিয়ে।

এবার, আপনি আপনার কম্পিউটারকে দিয়ে যাই করাতে চান না কেন, তার একটা নিজস্ব প্রক্রিয়া আছে। এই প্রক্রিয়ার প্রাথমিক ধাপ দুটো। এক, কী করাতে চান, সেটাকে যুক্তিশীল এবং বোধ্য রকমে পরপর লিখে ফেলা। এ

কাজেই আমাদের প্রয়োজন পড়ে কম্পিউটার ভাষা। সি যে ভাষাদের অন্যতম। এই সি ভাষা আমাদের কাছে বোধ্য, কারণ আমরা সেই ভাষা না-বুঝলে সেই ভাষায় নির্দেশ দেব কী করে? এবার ধাপ দুই, মানে, এই সি ভাষায় লেখা নির্দেশমালাকে কম্পিউটার-বোধ্য করে তোলা। যে কাজটা করে কম্পাইলার। সি থেকে সরাসরি মেশিন ল্যাংগুয়েজে নেমে প্রোগ্রামটার একটা মেশিন-ভাষায় মেশিন বোধ্য অনুবাদ তৈরি করাই কম্পাইলারের কাজ। এই প্রতিরূপটা আর আমাদের কাছে বোধ্য নয়, হওয়ার কোনো প্রয়োজনও নেই। তাই, একটা চালনীয় বা এক্সিকিউটেবল ফাইল এডিটর দিয়ে খুলে আমরা ওই সব বিচিত্র জিনিষ দেখতে পারি, ইটি এবং অ্যালিয়েন যে লিপিতে নিজেদের মধ্যে চিঠি-চালাচালি করে। ছিল একটা টেক্সট ফাইল, তাতে সি ভাষায় লেখা মানববোধ্য আদেশ। কম্পাইল করে সেটা থেকে পাচ্ছি একটা চালনীয় বা এক্সিকিউটেবল ফাইল। যাতে ভরা কম্পিউটারবোধ্য মেশিনভাষার আদেশমালা। সেই আদেশ কম্পিউটার বুঝতে পারে, তাই সেই ফাইল সরাসরি চালানো যায়, কমান্ড প্রম্পটে সেই ফাইলের নাম লিখে এন্টার মারা মানে কম্পিউটারকে জানানো, তুমি এই এক্সিকিউটেবল ফাইলে ভরা আদেশগুলো পালন করো, মানে, মোদ্দা কথাও তুমি এই ফাইলটা চালাও।

এই কম্পাইলার আছে হরেক ব্রান্ডের, নানা কোম্পানির। উইন্ডোজ জগতে মাইক্রোসফট বা বোরল্যান্ড ডাভিডি বা এখন ইনটেলের, এবং অন্য আরো অনেক আছে। এবং সবাই নিখাদ কম্পাইলারও নয়, তাতেও নানা ধাঁচের জটিলতা আছে। নানা বিশেষ বিশেষ কাজের বিশেষ বিশেষ ব্যবস্থা আছে। গ্নু-লিনাক্স জগতে আছে জিসিসি, যা একই সঙ্গে সি, সি প্লাস প্লাস, ফোর্ট্রান, অবজেক্টিভ সি, জাভা — যাবতীয় ভাষায় লেখা প্রোগ্রামই কম্পাইল করতে পারে। মানে আপনি যাকে কমান্ড বলে বুঝতে পারছেন কম্পিউটার ভাষা জানার দৌলতে তাকে কম্পিউটারের কাছেও বোধ্য করে তুলতে পারে, সেই কাজটা তাকে দিয়ে করিয়ে নেওয়ার ব্যবস্থা করতে পারে।

তাহলে, কম্পাইলার ঠিক কী করছে? কম্পাইল করা মানে কী? পরে আমরা এটায় বিশদ ভাবে আসব, আপাতত একটা আলগা আন্দাজ তৈরি করে রাখি। এই যে সি নামক কম্পিউটার ভাষায় লেখা আদেশে ভরা কোড ফাইলটা, মানে 'gadha.c', এটা তো বললাম, কম্পিউটারের কাছে অবোধ্য। কম্পাইলার এবার তাকে কম্পিউটার বোধ্য করে তুলবে, যাতে কম্পিউটার সেই আদেশগুলোকে বুঝতে পারে, তাই পালন করতে পারে। স্ক্রিনে ফুটিয়ে তুলতে পারে সেই চূড়ান্ত সমাদর, 'কিরে গাধা!!!'।

ধরুন এটা একটা গ্নু-লিনাক্স মেশিন, এবং আমাদের কম্পাইলার জিসিসি (gcc — **G**NU-**C**ompiler-**C**ollection)। এই অবস্থায়, 'gadha.c' ফাইলটা লেখা এবং সেভ করার পর, আমরা কমান্ড প্রম্পটে টাইপ করে জিসিসি-কে কমান্ড দেব, '`gcc -o gadha gadha.c`'। যার অর্থ, 'gcc' নামের কম্পাইলার ব্যবহার করে এই 'gadha.c' ফাইলটাকে কম্পাইল করো এবং কম্পাইল করে বানানো সেই চালনীয় বা এক্সিকিউটেবল ফাইলটার নাম দাও 'gadha'। এই 'gadha' হল সেই এক্সিকিউটেবল ফাইল বা প্রোগ্রাম। এবার কমান্ড প্রম্পটে গাধার নাম লিখে এন্টার মারলেই 'gadha' নামের প্রোগ্রামটা চলবে। মানে, 'gadha' নামের এক্সিকিউটেবল ফাইলে ভরা মেশিনবোধ্য আদেশগুলোকে পালন করবে। এই মেশিনবোধ্য আদেশগুলো এসেছিল 'gadha.c' নামক টেক্সটফাইলে ভরা মানববোধ্য আদেশগুলোকে জিসিসি নামের কম্পাইলার দিয়ে কম্পাইল করে।

এটা আমাদের অতিসরলীকৃত উদাহরণ। আদতে কমান্ডটা এত সহজ হয়না, সেখানে আরো অনেক কিছু থাকে। ফাইল লেখার সময় কোনো ভুল হলে সাবধান বাণী বা ওয়ার্নিং দিতে বলা থাকে কম্পাইলারকে। থাকে পোকা-ছাড়ানো বা ডিবাগিং সংক্রান্ত তথ্য ফাইলের মধ্যেই রেখে দেওয়ার আদেশ। থাকে অপ্টিমাইজেশন, মানে, কম্পিউটারের রসদের সঙ্গে নিজেকে মানিয়ে নেওয়ার আদেশ। ইত্যাদি। আবার, শুধু গ্নু-লিনাক্সে কেন, 'gadha.c' নামের কোড ফাইল কম্পাইল করা যাবে উইনডোজ প্লাটফর্মের কোনো সি-কম্পাইলার দিয়েও, সি নামক ভাষাটা তো একই। কিন্তু সেই কম্পাইল করার কমান্ডটা আলাদা হবে। এই একই কমান্ডে জিসিসি দিয়েই উইনডোজ প্ল্যাটফর্মেই কম্পাইল করা যায় — ডিজেজিপিপি নাম কম্পাইলারটার, ইম্যাক্স দিয়ে কোড লেখো এবং জিসিসি দিয়ে সি প্রোগ্রাম করো উইনডোজ-এর মত খাজা প্লাটফর্মেই। শুধু তখন প্রোগ্রাম ফাইলটার নাম হবে 'gadha' নয়, 'gadha.exe' — উইনডোজ-এর ওটাই নিয়ম। '.exe' নামক ফাইল-টাইটেলটা বোঝাচ্ছে যে এটা একটা চালনীয় বা এক্সিকিউটেবল ফাইল, মানে প্রোগ্রাম। সরাসরি একে চালানো যাবে, কমান্ড প্রম্পট থেকে বা ইঁদুরের পেট টিপে আদেশ দিয়ে। এবার যদি এই প্রোগ্রাম ফাইল, সে গ্নু-লিনাক্সের 'gadha' হোক বা উইনডোজের 'gadha.exe', কোনো

সরল কমান্ড এডিটর দিয়ে খোলেন, তাতে আবার দেখবেন সেই অবোধগম্যতা, কিন্নর গন্ধর্বদের প্রাইমারির সিলেবাসে অবশ্যপাঠ্য সেই মেশিন ল্যাংগুয়েজ। আমরা বুঝব না, কিন্তু মেশিন বুঝবে। এবার আপনি ওই প্রোগ্রামটাকে সরাসরি চালান, লিনাক্স হলে ./gadha, বা উইনডোজ হলে gadha.exe — আদেশের চেহারাটা এরকম আলাদা হল কেন আমরা পরে বুঝব — সরাসরি ঠিক তাই করবে যা আমরা তাকে করতে বলেছিলাম, মানে, সেই সমূহ সম্মাননায় সমৃদ্ধ লাইনটিকে ফুটিয়ে তুলবে স্ক্রিনে, 'কিরে গাধা!!!'। মানে, আমাদের কম্পাইলার মূল কোড ফাইলটাকে কম্পাইল করে কম্পিউটারবোধ্য করে তুলল। যাতে এবার নির্দেশমালাটাকে কম্পিউটার বুঝতে এবং পালন করতে পারল।

৭।। অ্যাপ্লিকেশনস

হার্ডওয়ার থেকে সফটওয়ার অব্দি কম্পিউটারের গোটা কাজের এলাকাটার স্তরবিন্যাস দেখিয়ে আমাদের দেওয়া সারণীর ৪ নম্বর স্তরে অপারেটিং সিস্টেম একা, আর ৫ নম্বর স্তরে তিনটে উপাদান কমান্ড ইন্টারপ্রিটার, এডিটর এবং কম্পাইলার। এখানেই শেষ মূল প্রোগ্রাম কাঠামো বা সিস্টেম প্রোগ্রামের এলাকা। এর পর শুরু প্রয়োগ বা অ্যাপ্লিকেশনের এলাকা। যে এলাকায় মেশিন ঘিরে বসে ব্যবহারকারীরা তাদের নানা প্রয়োজনের নানা প্রয়োগমূলক প্রোগ্রাম বা অ্যাপ্লিকেশন সফটওয়ার চালায়। এখানে আমরা তিনটে উদাহরণ দিয়েছি, অ্যাকাউন্টিং সফটওয়ার, রেলওয়ের টিকিট বন্টন ব্যবস্থা আর ইন্টারনেটে ওয়েব ব্রাউজিং। এরকম আরো বহু সফটওয়ার আছে। এটাই হল সেই স্তর যেখানে সচরাচর বেশিরভাগ ব্যবহারকারী বা ইউজার কাজ করে।

এই জায়গায় একটা কথা বলার আছে। উইনডোজ জগতে এই ব্যবহারকারী বা ইউজার এবং পরিচালক বা সুপারভাইজর — এই দুটো আলাদা জায়গা খুব একটা স্পষ্ট নয়। কিছুটা আছে উইনডোজ এক্সপিতে। আর গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স সিস্টেমেই এই ব্যাপারটা সর্বব্যাপী। এটাই এই মেশিনগুলোর নিরাপত্তার উৎস। কেন কোনো গ্নু-লিনাক্স সিস্টেমে কোনো ভাইরাস ইনফেকশন সম্ভব না, একদম কাঠামোগত ভাবেই, সেটা বুঝতে গেলে আমাদের এটা বুঝতে হবে। এটা নিয়ে বিশদ আলোচনা আসতে বহু দেরি আছে। সাত বা আট নম্বর দিনের আগে নয়। কারণ, তার আগে, ফাইল তথা ফাইলসিস্টেম তথা তার অধিকার এবং অনুমতির কাঠামো নিয়ে বহু কথা বলার আছে, যার উপর দাঁড়িয়ে আছে ইউনিক্স তথা যে কোনো গ্নু-লিনাক্স ব্যবস্থার গোটা কার্যধারাটা।

ইউনিক্স জগতে, সেটা ব্রান্ডনেম ব্যবসায়িক ইউনিক্স হোক, আর ওপন-সোর্স ইউনিক্স মানে গ্নু-লিনাক্স হোক, এই পরিচালক হল সর্বব্যাপী একচ্ছত্র ক্ষমতার অধিকারী, সিস্টেমের আভ্যন্তরীন প্রতিটি প্রত্যেকটি কিছুই তার ইচ্ছাধীন। এর নাম রুট। এবং ব্যবহারকারীরা ঠিক ততটুকু অধিকারই উপভোগ করে যতটুকু রুট তাকে অনুমতি দেয়। আমরা পরে ধীরে ধীরে এর সমস্ত জটিলতাগুলো বুঝতে পারব। এখন শুধু এটুকুই মাথায় থাক যে সাধারণ ব্যবহারকারী বা ইউজারের এলাকার সঙ্গে রুট বা সুপারভাইজরের অধিকারের একটা নাটকীয় তফাত আছে।

অপারেটিং সিস্টেমের স্তরের অন্য নাম কারনেল স্তর। এই কারনেল স্তর গোটাটাই সুপারভাইজরের এলাকা। সাধারণ ব্যবহারকারীর এখানে কোনোরকম কোনো প্রবেশাধিকারই নেই। তার আওতায় পড়ে এই অ্যাপ্লিকেশন সফটওয়ারগুলো ব্যবহার করা, তাও তার খুঁটিনাটি নিয়ন্ত্রকগুলো লেখা থাকে যে কনফিগারেশন ফাইলে, যে ফাইল অনুযায়ী অপারেটিং সিস্টেম এদের কাজ করার ব্যবস্থা করে দেয়, বিভিন্ন হার্ডওয়ার উপাদানগুলো এদের ব্যবহার করার সুযোগ করে দেয়, সেই কনফিগারেশন ফাইলগুলোও বদলানোর কোনো অধিকার থাকে না একজন সাধারণ ব্যবহারকারী বা ইউজারের। সাধারণ ব্যবহারকারী কমান্ড এডিটর দিয়ে প্রোগ্রাম লিখতে পারে, কম্পাইলার দিয়ে সেটা কম্পাইলও করতে পারে, কিন্তু কোনো সিস্টেম প্রোগ্রামে হাত দিতে পারে না। মানে সিস্টেম কাজ করার জন্যে যে প্রোগ্রামগুলো প্রয়োজন পড়ে। আমাদের টেবিল মোতাবেক বলতে গেলে, ৫ নম্বর স্তরে এসে কম্পাইলার এডিটর কমান্ড ইন্টারপ্রিটারদের নিয়ে সুপারভাইজর বা রুট-এর একচ্ছত্র অধিকারের এলাকা শেষ। এক থেকে পাঁচ অব্দি কারনেল মোড। কোনো ব্যবহারকারীর কোনো রকম স্বাধীনতা এই অব্দি বরাদ্দ নয়। কোনো সাধারণ ব্যবহারকারীর তৈরি করা কোনো প্রোগ্রাম সিস্টেম কোনো প্রোগ্রামে তথা কনফিগারেশনে তথা ফাইলে হাত দিতে পারেনা। কেন পারেনা এটা বোঝার জন্যে আমাদের ছয় নম্বর দিন অব্দি অপেক্ষা করতে হবে, যখন আমরা একটা

ফাইলের উপর মালিকানা তথা পড়ার/ লেখার/ চালানোর এই তিন রকম অধিকারের কাঠামোটা জানতে পারব। এটাই একটা গ্নু-লিনাক্স ব্যবস্থার সমস্ত নিরাপত্তার উৎস।

এর পর শুরু ইউজার বা ব্যবহারকারীদের এলাকা। বা শেল মোড। যারা বিভিন্ন অ্যাপ্লিকেশনকে প্রয়োগ করে বিভিন্ন কাজে। বা অকাজেও, যা তারা করতে চায়। লেখা থেকে মিউজিক বা ভিডিও এনকোডিং, গেমস থেকে থ্রি-ডি রেন্ডারিং, স্যাটেলাইট সুরভেইলান্স থেকে বিদ্যাচর্চার সর্বব্যাপী উন্মুক্ততা। নতুন প্রোগ্রাম লেখা থেকে শুরু করে পুরোনো প্রোগ্রামের চুল চিরে দেখা — যার অন্য নাম ডিবাগিং। এই প্রয়োগমূলক প্রোগ্রামের একটা বড় এলাকাকে আমরা আমাদের এই পাঠমালাতে আনবই না, যারা গুই বা গ্রাফিকাল-ইউজার-ইন্টারফেস বা ছবি-মাউসের জগতের অ্যাপ্লিকেশন। এই পাঠমালায় আমরা একটা গ্নু-লিনাক্স ব্যবস্থাকে শুধু তার কমান্ড মোডেই ধরার চেষ্টা করব।

আমরা আমাদের টেবিলে আলাদা আলাদা স্তরগুলোকে খুব স্পষ্ট ভাবে ভাগ করেছি। সবসময় অবশ্য সবগুলো স্তরকে এমন জল-অচল কামরায় আলাদা করা যায়না। যেমন রিয়াল-টাইম এমবেডেড সিস্টেমগুলো, ধরুন ক্রেডিট কার্ড বা সেলফোন ইত্যাদি। সেখান অপারেটিং সিস্টেম এবং অ্যাপ্লিকেশন এরকম স্পষ্ট ভাবে বিভক্তই নয়। বা এমনকি কোথায় ইউজার তথা অ্যাপ্লিকেশনের এলাকা শুরু এবং কোথায় সিস্টেম প্রোগ্রামের এলাকা শেষ তা নিয়েও চুলচেরা বিচারে অনেক তর্ক উঠতে পারে। কিন্তু সেগুলো আমাদের এই পাঠমালার আওতায় কোনোভাবেই পড়বে না।

অপারেটিং সিস্টেমের ভিত, এবং তার উপর দাঁড়িয়ে কম্পিউটারের গোটা কাজের কাঠামোটার সম্পর্কে একটা প্রাথমিক ধারণা হল আমাদের। এক নম্বর দিন শেষ হল। এরপর, দুই নম্বর দিন, আমরা সিপিইউ এবং তার চারপাশে ঘিরে থাকা নানা ভৌত উপাদানগুলোর সম্পর্ক নিয়ে প্রাথমিক একটা ধারণা তৈরি করব। তিন আর চার নম্বর দিনে আসবে হার্ডওয়ার এবং সফটওয়ারের দুটো নিকট দুনিয়াকে মিলিয়ে প্রাগইতিহাস থেকে ইতিহাসে কম্পিউটারের বিবর্তন। পাঁচ নম্বর দিন থেকে আমরা পুরোদমে একটা জ্যান্ত গ্নু-লিনাক্স সিস্টেমে ঢুকব।